A Framework for the Coordination of Complex Systems’ Development

by

Lars Taxén

Department of Computer and Information Science
Linköpings universitet
SE-581 83 Linköping, Sweden

Linköping 2003
Abstract

This study is about the coordination of complex systems’ development. A Framework has been designed and deployed by the author in the development practice of Ericsson, a major supplier of telecommunication systems on the global market. The main purpose of the study is to investigate the impacts on coordination from the Framework. The development projects are very large and subject to turbulent market conditions. Moreover, they have many participants (often several thousand), have tight time constraints and are distributed to many design centres all over the world. In these projects, coordination of the development is of crucial importance. The Framework is grounded in a tentative theory called the Activity Domain Theory, which in turn is based on the praxis philosophy. In this theory the interaction between the individual and her environment is mediated by signs. Coordination is conceived as a particular activity domain which provides coordination to the development projects. The coordination domain is gradually constructed by the actors in this domain by iteratively refining a conceptual model, a process model, a transition model, a stabilizing core and information system support. In this process individual knowledge, shared meaning and organizational artefacts evolve in a dialectical manner. The Framework has been introduced in the Ericsson development practice over a period of more than ten years. Between 1999 and 2002 approximately 140 main projects and sub-projects at Ericsson have been impacted by the Framework. These projects were distributed to more than 20 different development units around the world and were carried out in a fiercely turbulent environment. The findings indicate that the Framework has had a profound impact on the coordination of the development of the most complex nodes in the 3rd generation of mobile systems. The knowledge contributions include an account for the history of the Framework at Ericsson and an identification of elements which contribute to successful outcomes of development projects.

This study has been supported by the Industry Research School in Applied IT and Software Engineering at Linköping University, the Swedish National Board for Industrial and Technical Development, NUTEK, project P10518-1 and Ericsson AB.
Preface

The road to wisdom?—Well, it's plain and simple to express:

Err
and err
and err again
but less
and less
and less. (Piet Hein, inventor, poet)

This dissertation is the result of an enduring educational journey which has been swaying between the attraction points of a deep interest in philosophy, my work at the Ericsson company and my research at Linköping University. After working more than 30 years at Ericsson I finally got my act together and switched to a new track. This is a decision I have never regretted.

Most dissertations come early in life as the start of a long career. However, for me rather the opposite is true. I am writing up my experience of a long career. Thus, I have allowed myself to be generous in what material I have included in the dissertation. On the surface, issues like the philosophical question of what knowledge is or how the cognitive system of humans conceives signs, may seem rather loosely related to, for example, requirement management. However, it is my firm conviction that it is not until we understand and reflect on these seemingly non-related issues that we can truly address the extraordinary intricacies in complex systems’ development.

For some peculiar reason this educational journey seems to be intrinsically related to actual trains. Like some trains nowadays, I started late. Sometimes the chaos in my head was equally matched by the chaos at the train station. Nothing happened, no information was available of what is going on, I had no idea when I would arrive at the end station. However, most of the time the travelling has been pure joy. Each time I boarded a train in Stockholm to go to any one of those ‘köpings’ (Lin-, Norr- or Jön-) it became all too evident what I missed for such a long time: to just sit down, take it easy, look out the window and reflect on what is going on all around.

This journey would not have left the first station if it wasn’t for professor Sture Hägglund, who looked at my ideas over a cup of coffee at the central station in Stockholm, then looked at me, rubbed his chin and said: “Well, this might be something...”. He put me in contact with my supervisor Bengt Lennartson who has been my guide throughout this journey. His patient and thorough criticism and encouragement have kept me on track whenever I was about to get lost at some interesting train stop. Hail to you, Bengt!

The same goes for my additional supervisors, Roland Ekinge at Whirlpool and professor Christian Berggren. Roland, always asking those provoking questions which (hopefully) taught me to challenge my own preconceptions and think instead of becoming defensive.
Christian, with whom I often discussed my shaky thoughts on a shaky X2000 to or from Linköping.

Also, this journey would not have come to an end without Sören Ohlsson who always welcomed me at the Ericsson station. His continual interest in my results, his long experience from Ericsson and his encouragement has been invaluable to me.

Perhaps the trickiest part of my journey has been to get on track at the philosophical station. Oh, how easy it is to get lost in a complete confusion here! “Philosophy is like trying to open a safe with a combination lock: each little adjustment of the dials seems to achieve nothing, only when everything is in place the door does open.” (Ludwig Wittgenstein). For their help in break open this safe I am deeply indebted to two people: professor Göran Goldkuhl and Johan Schubert (guess where I met Johan: on a train...). Göran, whose graduate courses and seminars together with his students have provided me with countless occasions for discussion and reflection. Johan, with his long experience as an opponent and supervisor, was always there to inspire, read and gently point out the flaws in my thinking. My special and warm thanks also go to all of you who took the time to read and comment on my dissertation and in particular Pär Ågerfalk and Pär Carlshamre who did so on several occasions.

In the summer of 2002 I was dismissed from Ericsson which of course was a dramatic event in my life and a severe blow to my self-confidence. To all of you at Ericsson, who expressed those simple but dear words of sympathy and consolidation, I am truly grateful. In particular I am grateful to Lars-Göran Andersson who always popped by, listened and just cared. Thank you, Lars-Göran, for being such a good colleague.

When I first came to the campus in Linköping in 1998, I was completely lost both geographically and mentally. Where is the department, how do you get started, how do I behave, whom do I ask and so on. Numerous were those seminars when I looked around an empty classroom at 1 p.m. before I realized that 1 p.m. means 1.15 p.m. at the university... For gently guiding me throughout all the labyrinths of the scientific workplace I am truly grateful to Lillemor Wallgren who has been accompanying me from the start all through to the defence and Britt-Inger Karlsson for your helpful assistance.

So, when this journey is approaching its destination, my final thoughts go to you who have been with me all the time: Tina for your ceaseless love and endurance, my children Kerstin and Gustav for your everlasting encouragement. And of course to Eva, my daughter-in-law and Anna, my granddaughter, now three years old, who joined us on the train halfway through. Perhaps the essence of this trip is best expressed by her. When looking at a coloured version of the work package dependencies in the mobile switching centre node (Figure 2, p. 3), she thought it looked like “square clouds over an island in the sea”. Perhaps we should think about work packages as square clouds. Or as my favourite philosopher, Karel Kosík, expressed it: “Familiarity is an obstacle to knowledge”.

Tullinge, February 2003

Lars
## Contents

### PART I - INTRODUCTION AND RESEARCH DESIGN

1 INTRODUCTION .................................................................................................... 1
   1.1 The study field............................................................................................... 1
   1.2 Research aims ............................................................................................. 4
      1.2.1 Purpose and main research question ..................................................... 5
      1.2.2 Coordination .......................................................................................... 6
      1.2.3 A remark about the thesis in the thesis ................................................ 7
      1.2.4 Complexity ............................................................................................ 8
      1.2.5 System .................................................................................................. 9
      1.2.6 Information systems ............................................................................. 9
      1.2.7 Model ................................................................................................... 10
      1.2.8 Project .................................................................................................. 10
      1.2.9 The empirical data ............................................................................... 10
   1.3 The Framework and its grounding theoretical perspective ...................... 11
      1.3.1 The Framework...................................................................................... 11
      1.3.2 The Activity Domain Theory .................................................................. 12
   1.4 Expected research results ........................................................................... 13
      1.4.1 Detailed research questions ................................................................... 14
      1.4.2 Additional knowledge contributions ..................................................... 15
      1.4.3 Characterizing the knowledge contributions ......................................... 16
   1.5 Related work ............................................................................................... 16
   1.6 My personal background .......................................................................... 19
   1.7 Delimitations ............................................................................................... 20
   1.8 Stakeholders ............................................................................................... 21
   1.9 Structure of the dissertation ....................................................................... 23
      1.9.1 Reading guidelines ............................................................................... 24
   2 RESEARCH DESIGN AND REALIZATION .................................................. 25
      2.1 Positions taken .......................................................................................... 25
         2.1.1 An interpretative research tradition founded in praxis ....................... 25
         2.1.2 The roles of theories ............................................................................. 26
         2.1.3 A single-case, multiple-units of analysis study ................................... 28
         2.1.4 Action research .................................................................................... 30
         2.1.5 Validity, reliability and relevance ......................................................... 32
         2.1.6 Generalizations from interpretative research ..................................... 33
         2.1.7 The artefact in science ......................................................................... 34
      2.2 Realization ................................................................................................. 35
         2.2.1 Coordination domains at Ericsson ...................................................... 35
         2.2.2 Interviews ............................................................................................ 36
         2.2.3 Data triangulation ............................................................................... 40
         2.2.4 Grounding in industrial settings .......................................................... 41
         2.2.5 Industrial grounding internally at Ericsson ......................................... 42
         2.2.6 Industrial grounding externally outside Ericsson ................................ 43
         2.2.7 Research processes, strategies and methods ....................................... 43
4.1.6 Domain transition ................................................................. 102
4.1.7 The stabilizing core.............................................................. 103
4.1.8 Tool usage........................................................................... 103
4.2 Summary.................................................................................. 104

5 THE STRUCTURE OF THE FRAMEWORK.............................................. 105
5.1 The Framework elements and how they are related.................. 105
5.1.1 The Conceptual Model.............................................................. 108
5.1.2 The Process Model................................................................. 108
5.1.3 The Transition Model .............................................................. 110
5.1.4 The Information System .......................................................... 112
5.1.5 The Stabilizing Core ............................................................... 114
5.1.6 The Domain Construction Strategy ........................................ 114
5.2 Summary.................................................................................. 115

PART III - RESULTS FROM THE ERICSSON PRACTICE

6 THE HISTORY OF THE FRAMEWORK................................................. 119
6.1 A pattern emerges (1990-1995).................................................... 119
6.1.1 Concept elaboration................................................................. 120
6.1.2 The system design environments............................................. 121
6.1.3 Information management........................................................ 125
6.1.4 Information systems............................................................... 126
6.1.5 Towards the Framework .......................................................... 127
6.2 Coordinating incremental development (1996-1997).................... 128
6.2.1 The Incremental Development Method Package....................... 128
6.2.2 The Information System enters the stage................................. 130
6.2.3 The pilot projects................................................................. 130
6.3 Termination and despair (1998)..................................................... 134
6.4 Renewal in conflicts - the pioneering S-domain (late 1998-1999).... 137
6.4.1 The Beamon project............................................................... 137
6.4.2 Fighting problems................................................................. 139
6.4.3 Escalation............................................................................... 141
6.4.4 Conflicts............................................................................... 142
6.5 Other practices catch on (2000)...................................................... 143
6.5.1 The L-domain in Linköping..................................................... 143
6.5.2 The A-domain in Aachen....................................................... 144
6.6 A central concern - the C-domain (late 2000 - 2001).................... 145
6.6.1 The reference system.............................................................. 146
6.6.2 The Core Network starts using eMatrix................................... 147
6.6.3 Corporate on stage............................................................... 148
6.7 From federalism to monarchy (2002)............................................ 148
6.8 Analysis of the Framework trajectory........................................ 149

7 FRAMEWORK CONSEQUENCES................................................... 155
7.1 Penetration into the Ericsson development practice..................... 155
7.2 The structure of coordination domains....................................... 155
7.2.1 Coordination situations ................................................................. 155
7.2.2 Actor roles and groups ............................................................... 158
7.3 Consequences .................................................................................. 159
  7.3.1 Sine qua non ............................................................................ 160
  7.3.2 Evolution .................................................................................. 161
  7.3.3 Orientation ............................................................................... 165
  7.3.4 Separation of concerns ............................................................ 169
  7.3.5 Centralism – decentralism ....................................................... 171
  7.3.6 Deployment ............................................................................. 173
  7.3.7 Data reliability ......................................................................... 174
  7.3.8 The Information System – eMatrix ........................................ 177
  7.3.9 Efficiency ................................................................................ 180
  7.3.10 Managerial issues ................................................................. 183
7.4 Information System Architectures .................................................... 186
  7.4.1 The traditional architecture ..................................................... 186
  7.4.2 The alternative architecture .................................................... 188
  7.4.3 The corporate connection ....................................................... 189
7.5 The construction of shared meaning ................................................. 192
  7.5.1 The agony of achieving shared meaning - an example ............. 192
8 REINFORCEMENT RODS..................................................................... 199
  8.1 The experiences of PM1 ............................................................. 199
    8.1.1 Open coding ........................................................................ 200
    8.1.2 Axial coding ......................................................................... 201
  8.2 Selective coding - reinforcements related to the Framework .......... 204
    8.2.1 Evolution ............................................................................. 205
    8.2.2 Orientation .......................................................................... 206
    8.2.3 Federalism ........................................................................... 207
    8.2.4 Balance ................................................................................ 208
    8.2.5 Isomorphism ........................................................................ 209
    8.2.6 Shared meaning .................................................................... 210
  8.3 Selective coding - other reinforcements ........................................ 211
    8.3.1 Participation ......................................................................... 211
    8.3.2 Focus ..................................................................................... 211
    8.3.3 Turning points ....................................................................... 212
PART IV - DISCUSSION & CONCLUSIONS
9 DISCUSSION OF THE FRAMEWORK APPROACH ............................. 215
  9.1 The Framework and its elements .................................................. 215
    9.1.1 Reflections on the Framework as a whole ............................... 215
    9.1.2 The Conceptual Model .......................................................... 216
    9.1.3 The Process Model ............................................................... 217
    9.1.4 The Transition Model ......................................................... 218
    9.1.5 The Stabilizing Core ............................................................ 219
    9.1.6 The Information System ....................................................... 220
    9.1.7 The Domain Construction Strategy ....................................... 221
  9.2 Coordination of development projects ......................................... 223
9.2.1 Impacts on coordination................................................................. 223
9.2.2 Balancing central control and local autonomy.............................. 224
9.2.3 Development of large, global information systems.......................... 226
9.2.4 One information system supporting coordination........................... 226
9.2.5 Construction of shared meaning...................................................... 227
9.2.6 Management support - does not have to be top management........... 227
9.3 The Activity Domain Theory............................................................. 228
9.4 Generalizing the results..................................................................... 228

10 CONCLUSIONS ..................................................................................... 231
10.1 Recapitulation of research questions.................................................. 231
10.2 Conclusions concerning the impacts on coordination......................... 231
10.2.1 Profound impacts on coordination from the Framework intervention ... 231
10.2.2 Balancing central control and local autonomy – a key issue.............. 232
10.2.3 Evolutionary development of global information systems – a reality ... 232
10.2.4 Coordination domain construction in three phases......................... 233
10.2.5 One information system for coordination provides major advantages ... 233
10.2.6 Shared meaning and coordination support evolve simultaneously........ 234
10.3 Conclusions concerning the reinforcement rods................................. 234
10.4 Conclusions from the history of the Framework................................. 235
10.5 Additional conclusions...................................................................... 235
10.6 Research design and method evaluation.............................................. 236
10.7 Further areas of research................................................................. 238
10.7.1 Investigating the application domain of the Framework................... 238
10.7.2 Evolutionary information system development.............................. 238
10.7.3 Consolidate reinforcements............................................................ 238
10.7.4 The optimum balance between control and autonomy.................... 238
10.7.5 A systems development method based on the Framework................ 239
10.7.6 Product life cycle management – towards a new foundation............. 239
10.7.7 Organizational learning................................................................. 239
10.8 Concluding remarks......................................................................... 240
10.8.1 The quality criteria of the dissertation............................................. 240
10.8.2 Limitations of the study................................................................. 242
10.8.3 Significance for Ericsson and other industries............................... 242
10.8.4 The industry Ph.D. student – some reflections................................. 243
10.8.5 Summary of the knowledge contributions....................................... 244

11 REFERENCES ....................................................................................... 245
11.1 From the literature........................................................................... 245
11.2 Ericsson references........................................................................... 252
11.3 Ericsson references (Taxén)............................................................... 253
11.4 World wide web.............................................................................. 254

APPENDIX 1: GROUNDING OF EFFECT CATEGORIES ............................... 255
INDEX ...................................................................................................... 265
List of Figures

Figure 1. An example of a telecommunication network (after Lindman, 2000) ......................... 2
Figure 2. Dependencies between development tasks in a network node .................................. 3
Figure 3. The same items in the coordination domain and the usage domain ............................ 7
Figure 4. The Framework and its elements .............................................................................. 11
Figure 5. How the knowledge contributions are derived ......................................................... 15
Figure 6. Considered success factors ....................................................................................... 20
Figure 7. The role of the dissertation ...................................................................................... 23
Figure 8. The outline of the dissertation .................................................................................. 28
Figure 9. Science, engineering and design (adapted after Hendersen, 1998) ............................ 35
Figure 10. Research processes in the study ............................................................................ 45
Figure 11. Conceptual model used in the analysis .................................................................... 46
Figure 12. An example of data analysis ................................................................................... 47
Figure 13. The line of reasoning ............................................................................................. 52
Figure 14. Signification as a relation ....................................................................................... 63
Figure 15. A layered model for cognitive systems (adapted after Gärdenfors, 2000a) .......... 67
Figure 16. A stratified interaction model ................................................................................ 68
Figure 17. Image-schema for “over” and “under” (adapted after Gärdenfors, 1992, p. 99) ...... 71
Figure 18. Illustration of a focal situation .............................................................................. 76
Figure 19. Some notions regarding contexts ........................................................................... 77
Figure 20. Adaptability of an organization as a function of enterprise control ...................... 82
Figure 21. The structure of activity domains ........................................................................... 85
Figure 22. The structure of the activity .................................................................................... 89
Figure 23. The interaction between the coordination and usage domains .............................. 98
Figure 24. The elements of the Framework ............................................................................ 107
Figure 25. Conceptual map of the Framework and its elements .............................................. 107
Figure 26. Alternative process models ................................................................................... 108
Figure 27. A Change Request Process Model in the IFD notation ......................................... 110
Figure 28. Interacting activity domains .................................................................................. 110
Figure 29. The Specification Based Data Model ...................................................................... 111
Figure 30. The Transition Model ........................................................................................... 111
Figure 31. The implementation of the models in the Information System ............................... 113
Figure 32. Results from the Domain Construction Strategy .................................................. 115
Figure 33. A layered model for cognitive systems (adapted after Gärdenfors, 2000a) .......... 119
Figure 34. Process components and SBDM ............................................................................. 123
Figure 35. Information Flow Diagrams LTX-1994-07-06, p. 6 ............................................. 123
Figure 36. Information Flow Diagram: multi-chip module design (ERI-1995-03-18) .............. 124
Figure 37. First version of the Conceptual Model (late 1996) ............................................... 129
Figure 38. The first version of the conceptual model supported by the Conceptual Model (late 1997) ........................................................................................................ 131
Figure 39. The Information System in the Framework, example (1997) ................................ 132
Figure 40. The Process Model in the Framework, example (1997) ........................................ 132
Figure 41. The Conceptual Model (late 1999) ........................................................................ 139
Figure 42. The Conceptual Model in the A-domain (2000) .................................................... 145
Figure 43. The trajectory of the Framework .......................................................................... 150
Figure 44. The Incremental Development Model .................................................................... 156
Figure 45. Coordination situations in the S-domain early 2000 ............................................. 156
Figure 46. Grounding of effect categories ............................................................................. 159
Figure 47. An anatomy ........................................................................................................... 166
Figure 48. Focal change .......................................................................................................... 167
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>49</td>
<td>Report from eMatrix showing dependencies</td>
<td>168</td>
</tr>
<tr>
<td>50</td>
<td>Dependency matrix, example</td>
<td>179</td>
</tr>
<tr>
<td>51</td>
<td>Traditional IS architecture</td>
<td>187</td>
</tr>
<tr>
<td>52</td>
<td>The scope of the eMatrix applications</td>
<td>188</td>
</tr>
<tr>
<td>53</td>
<td>The scope of the C-PDM system</td>
<td>190</td>
</tr>
<tr>
<td>54</td>
<td>Conceptual model - requirement focus - S-domain</td>
<td>193</td>
</tr>
<tr>
<td>55</td>
<td>Reinforcement rods from different projects</td>
<td>199</td>
</tr>
<tr>
<td>56</td>
<td>Different ways of using the Transition Model</td>
<td>219</td>
</tr>
<tr>
<td>57</td>
<td>The deployment phases</td>
<td>223</td>
</tr>
<tr>
<td>58</td>
<td>Development method</td>
<td>239</td>
</tr>
</tbody>
</table>
List of Tables

Table 1. Research questions ..................................................................................................... . 15
Table 2. Stakeholder interests.................................................................................................. .. 24
Table 3. Praxis in the Walsham classification (adapted after Walsham, 1995) ........................ 26
Table 4. Research questions ..................................................................................................... . 26
Table 5. Interviewee role and characterization......................................................................... 37
Table 6. Interview questions for research question RQ3 ........................................................... 38
Table 7. Interview questions for research question RQ2 ........................................................... 39
Table 8. Distribution of interviewees for research question RQ2 ............................................. 39
Table 9. Contributions – not peer reviewed .............................................................................. 41
Table 10. Publications – not peer reviewed ............................................................................ 42
Table 11. Internal grounding at Ericsson .................................................................................... 42
Table 12. Industrial reference visits ......................................................................................... 43
Table 13. Summary of the research design ............................................................................. 48
Table 14. Comparison between different approaches ............................................................... 92
Table 15. Roles, groups and access rights in the S-domain ......................................................... 158
Table 16. Consequence category – Sine qua non ..................................................................... 160
Table 17. Consequence category – Evolution .......................................................................... 161
Table 18. Consequence category – Orientation ...................................................................... 165
Table 19. Consequence category – Separation of concerns ...................................................... 169
Table 20. Consequence category – Centralism - decentralism .................................................. 171
Table 21. Consequence category – Deployment ...................................................................... 173
Table 22. Consequence category – Data reliability .................................................................. 174
Table 23. Consequence category – The Information System (eMatrix) ..................................... 177
Table 24. Consequence category – Efficiency ........................................................................ 180
Table 25. Consequence category – Managerial issues .............................................................. 183
Table 26. Consequence category – Information System Architectures .................................... 186
Table 27. Consequence category – Construction of shared meaning ....................................... 192
Table 28. Reinforcement rod – Evolution .............................................................................. 205
Table 29. Reinforcement rod – Orientation ............................................................................ 206
Table 30. Reinforcement rod – Federalism ............................................................................. 207
Table 31. Reinforcement rod – Balance .................................................................................. 208
Table 32. Reinforcement rod – Isomorphism ......................................................................... 209
Table 33. Reinforcement rod – Shared meaning ..................................................................... 210
Table 34. Layers in the Conceptual Model ............................................................................. 217
Table 35. Research questions and knowledge contributions .................................................... 231
Table 36. The grounding of effect categories ......................................................................... 255
PART I - INTRODUCTION AND RESEARCH DESIGN

In Chapter 1 Introduction the background of the dissertation and the study field at Ericsson are presented to the reader. The purpose and the main research question are given followed by a short description of the Framework and the Activity Domain Theory. Detailed research questions and expected knowledge contributions are described as well as additional knowledge contributions. Some related works and delimitations of the study are discussed. A disposition and reading guidelines close the first chapter.

In Chapter 2 Research Design and Realization the research design and realization are described. Positions taken regarding the type of research, the role of theories, the unit of analysis and action research are discussed as well as issues regarding validity, reliability, relevance and generalization of the results. The empirical sources and the grounding of the results are discussed. Strategies and method’s choices to obtain the expected knowledge contributions are accounted for. Restrictions concerning the data are given.
1 Introduction

Ideas thus made up of several simple ones put together, I call complex; such as beauty, gratitude, a man, an army, the universe. – Locke.

- Hello, hello, is there someone listening out there?
- Hi, I’m base station URK, who are you?
- I’m mobile 012-345678 and I belong to Lars.
- OK mobile, I’ll check if Lars has paid his bills and hasn’t stolen you. Please wait a moment.....OK fine, you are in and your position right now is cell ABC. You have been allocated channel HMRPF. I’ll keep an eye, sorry ear, on you.

- Hey mobile 012-345678, this is base station URK calling. I see that you are now leaving my control. I’ll hand you over to my colleague, base station OUCH. Here you go!
- Allo, allo mobile 012-345678, this is base station OUCH. You are now entering my area of command. You are allocated channel 4711, just relax....
- Thanks, base station OUCH, I know I’m in good hands...

- Mobile 012-345678, are you still there? This is base station OUCH. Someone is trying to call you!
- Who?
- It’s mobile 876-54321 belonging to Bengt. Please wake Lars up!
- OK, I’ll do that. Riinng!!!........OK, so now he’s awake, he has lifted the receiver, sorry pressed the receiver button.
- OK mobile, I’ll connect you after I started charging Bengt, I’d love to see his face when he gets the bill........ Done! Now we can relax while these guys are chatting. I wish I could understand what they are talking about...

Millions of “conversations” like this between mobile telephones and base stations go on every hour without our noticing it. In just a few years our daily lives have changed profoundly due to mobile telephony and the Internet. What was previously just fiction in novels and cartoons has now become routine for many people around the world. Just by pressing a couple of buttons we can contact each other almost wherever we happen to be for the moment. On the surface, these actions are quite simple but the underlying systems making this happen are very complex. This study is about the coordination of the development of such systems.

1.1 The study field

The ensemble of telecommunication systems has been called the world’s largest machine. It consists of a network of interacting nodes, each of which performs some kind of utility in
the overall network like keeping track of the position of the mobile, providing charging functions, supplying data about the mobile owner, etc. (see Figure 1.).

The interactions, as illustrated in the “conversation” above, are carried out according to complicated protocols with demanding requirements on performance, speech quality, etc. Other interactions are taking place within a node, for example a piece of software interacting with a piece of hardware such as a processor. On top of this there are interactions with various actors such as those performing maintenance and surveillance tasks.

Many different technologies are utilized in a telecommunication system. Radio technology is used to connect the mobile to the base stations. Base stations are the radio antennas visible along roads, on house roofs, etc. Software technology is used for various control purposes and for providing services like for example SMS (Small Message Service). Hardware technology is used in integrated circuits in the mobile, in processors and in switching equipment, etc. Optical technology is used in the cables connecting the nodes and in the lasers emitting the light pulses carrying the information. Mechanical technology is used in cabinets and magazines housing the equipment. Chemical technology is used in the design of the mobile hulls, in printed circuit boards, in wires, etc.

The network is in a state of constant evolution. New nodes are added and obsolete ones removed. New types of services are implemented and the capacity of the network is increased. For example, the now ongoing introduction of the 3rd generation of mobile systems will enable the transmission of video to and from the mobile. All the time, the network has to be monitored. For example, if a node stops working or a cable is cut off somewhere, a reconfiguration must be performed. In addition to this there is a legacy of existing equipment and networks which must be considered when making changes in the network. Several large customers, mainly the previously government controlled telecommunication departments, have been investing large amounts in their networks for a long time. Their equip-
INTRODUCTION

ment, which has a lifetime of several decades must live side by side with new technologies and new networks owned by other companies.

The character of telecommunication systems implies that the development of these systems is a complex task as well. The sheer size of the development task is enormous in terms of sub-products, developers, relationships, distribution, etc. A project developing a 3rd generation of mobile systems may take more than a year to execute and involve several thousand actors in different roles all over the world. This work is carried out at development sites which have a certain autonomy to structure their own way of working. During the development several hundred development steps, which may be regarded as mini-projects, must be coordinated.

To illustrate this complexity, the dependencies in one node in the mobile network (the Mobile Switching Centre node) are shown in Figure 2:

![Figure 2. Dependencies between development tasks in a network node](image)

Each white box represents a contribution to the overall functionality. The arrows between the boxes indicate dependencies among the development tasks, starting with the most basic task at the top of the figure. The white arrows mark the content and date for a particular integration and verification of a number of development tasks. The ‘bubbles’ represent basic services in the node like registration of the location of the mobile, calling to the mobile, answering a mobile call, etc. In most cases the functionality is provided by software. The
figure, which is called ‘anatomy’\(^1\) illustrates in principle the integration of a large software development project in which the number of lines of codes may be in the order of millions.

Furthermore, since there are a number of different technologies involved, very heterogeneous skills are needed. For example, the task of designing a mobile telephone antenna is very different from writing a software program to control a switch. Still, these and lots of other skills are needed to implement the whole system.

Since the projects are distributed over development organizations all over the world, the manner in which development is carried out differs from site to site. Local traditions and cultures have evolved over time. This applies to processes, tools, different ways of thinking, local markets conditions, etc. Still, there has to be some commonality in the project in order to put all the pieces together.

In the telecommunications market there is a fierce competition among systems providers like Ericsson, Nokia and others. Moreover, this market is changing rapidly, mainly due to two forces: the deregulation with the entering of many new operators which leads to more competition, and the proliferation of new technology such as mobile communications, intelligent networks, the Internet, etc. As a consequence, the suppliers have to be more reactive and flexible to the market needs, which means shorter lead times and an agile requirement management process.

Furthermore, the operators have currently invested heavily in licences and networks for the new generation mobile systems. So far the incomes have been less than expected which caused the market to more or less collapse during 2002. This has had a tremendous impact on development plans, organizational structure and employees in the organization. At Ericsson the staff was reduced from 107,000 employees worldwide to around 60,000 over a period of two years.

In order to meet the changes in markets and technologies, frequent re-organizations are carried out, including company outsourcing and new partnership constellations. For example, much of the previous in-house production of hardware equipment like printed circuit boards has in a few years been outsourced to external companies specializing in this type of business.

\(1.2\) Research aims

All the circumstances discussed above create a high level of stress on the capabilities of actors working in this area. In particular the one capability is affected: the coordination of the development task. A consorted action presumes that the actors apprehend ‘coordination’ in a similar way. However, the circumstances described above make it extremely difficult to evolve and maintain a shared meaning of coordination. This is even more difficult since product development organizations are used to focus on technical artefacts and tend to disregard social aspects such as how to achieve a shared meaning.

\(^1\) So called because the anatomy shows how the system is ‘coming alive’ from the base to the total functionality.
INTRODUCTION

Due to the turbulence and tough conditions on the market, a more or less constant re-planning must be performed which also affects the coordination. If the coordination fails for some reason, it is a threat to the entire project and business opportunities may be lost. The product may be delayed or perhaps not delivered at all to the customer, the delivery may contain erroneous or missing parts, etc.

The problem of coordination has been a subject for reflection and experimenting in my professional life for many years. The background is among other things my participation in both successful and unsuccessful projects. Gradually, starting in the early 1990s, I began to conceptualize and implement a Framework to support this type of coordination. The Framework has grown from a vague idea to a concrete artefact which is used in several development projects at Ericsson, including the development of the 3rd generation of mobile systems. It is from this background my research interest originates.

1.2.1 Purpose and main research question

The main purpose and the main research question of this study are formulated as follows:

_When developing complex systems subject to changing presumptions, the coordination of the development tasks is a crucial activity. A Framework has been developed and deployed in the development organization at Ericsson with the intention of supporting the coordination. The purpose of this study is to develop knowledge of the impacts of the Framework on the coordination task. The main research question is: What are the impacts on coordination from the Framework?_

In order to answer the main research question I will describe the Framework, reconstruct the history of it at Ericsson and analyse the overall consequences of its intervention, not only on coordination. Furthermore, I will identify a set of elements in the Ericsson development practice which contribute to the successful outcomes of projects. I will analyse how such elements are related to the Framework.

In addition to the main purpose there are some related purposes in the study. The Framework is grounded in a tentative theory called Activity Domain Theory. In this theory both individual / subjective as well as social / objective aspects are considered. I claim that this is necessary in order to cope with the complexity of the coordination task considered in this study. In Section 3.5 some alternative theories are discussed. However, their capacity to provide a basis on which operational coordination support can be erected appears to be limited. Thus, a first additional purpose is

_To evaluate the capacity of the Activity Domain Theory to inform the construction of a Framework for coordinating complex systems’ development._

---

2. I will mainly use ‘we’ throughout the study. However, in some parts the more personal ‘I’ will be used.
3. The world ‘Framework’ with a capital F refers to the Framework for coordination which is in focus for this study.
4. By ‘purpose’ I mean “expected result of an action”.

In Chapter 4 *Articulating Coordination* I suggest an elaborated conception of coordination as a form of praxis where actors are producing coordination for development projects. The reason for this is that I find the existing definitions of coordination in the literature insufficient for the operational purposes considered in this study. Thus, a second additional purpose is

> to evaluate the proposed conception of coordination with respect to the coordination of complex systems’ development.

Accordingly, I claim that the purpose and the main research question in this study are relevant from both a practical and theoretical point of view.

1.2.2 Coordination

I will ground my conception of coordination in the definition given by Malone & Crowston (1994):

> “Coordination is managing dependencies between activities” (Malone & Crowston, 1994, p. 90)

However, this definition needs to be further articulated. To begin with we must define *what* is being coordinated. In a coordination context, only phenomena related to coordination are important. All other phenomena associated with the entirety of the development activity will be subdued. For example, in the coordination context there is no difference between coordinating the development of a software or a hardware artefact. In other contexts though, the difference between them is vital. I will call the phenomena salient in the coordination context *coordination items*. These items are related to each other, they are acted upon in a certain order and they may appear differently in contexts outside the coordination context. Furthermore, since development practice is subject to never-ending changes, the content and structure of the coordination context must also change. This means that there is a need to continuously evolve and articulate this context and how the actors perceive it.

Elements which are subject to coordination in the development practice of Ericsson are typically

- requirements,
- products and documents,
- engineering change orders (called Change Requests at Ericsson),
- baselines and milestones in the project,
- intermediate development steps (called increments or work packages at Ericsson),
- test cases by which a product is tested for compliance with requirements,
- trouble reports indicating errors in the system, etc.

---

5. For example, in hardware integrated circuit design, you need to consider power consumption of a component which is not of primary interest in software design.
The next step in the articulation is to define what aspects are important when coordinating the development of complex systems. Some examples of such aspects are:

- support for incremental development,
- shared meaning concerning coordination,
- traceability between coordination items,
- transparency in the coordination context,
- dependencies between coordination items,
- global access to coordination information,
- project planning and monitoring,
- controlling changes and error correction,
- time to perform repetitive coordination tasks,
- number of information systems needed for coordination,
- number of interfaces needed between information systems, etc.

This means that we will include both artefacts like information systems and social issues like shared meaning in our articulation of ‘coordination’. Thus I will view coordination as something which is elaborated by a group of actors working together in an organized work setting which I will call a coordination domain. The coordination domain provides coordination to actors in the development project. This is another work setting which I will call the usage domain (see Figure 3).

1.2.3 A remark about the thesis in the thesis

I have avoided the explicit formulation of a thesis for the study because of the methodological difficulties in validating this. Any thesis related to the purpose of the study must in my opinion contain some value statement. For example, it would have been possible to formulate a thesis in the following way:

“The Framework improves the coordination of the development of complex systems when used during the development in comparison with not using it”
However, a key issue is how to define ‘improved’ and then evaluate the findings. Ultimately, ‘improved’ relates to the performance of the organization as such, which in my case is Ericsson. This performance can be evaluated along criteria like the cost of developing systems, development lead times, long- and short-term financial results, etc. These criteria must in turn be related to aspects of coordination, such as those given Section 1.2.2 Coordination above. In addition to these evaluations, the costs related to coordination must be evaluated. This cost includes not only the cost of the hardware and software needed but also the cost of discussions, conflict solutions, etc.

Thus, the effort to come up with a well-grounded answer to the thesis above would be an overwhelming task, if possible at all. I have therefore chosen to develop knowledge of the consequences of the Framework in relation to coordination without taking any normative standpoint. The intention of the Framework is, of course, that it will have positive effects on the coordination. However, I leave to the reader to be the final judge of the overall value of using the Framework in the coordination of complex systems’ development.

1.2.4 Complexity

A general definition of ‘complex’ may be “Consisting of interconnected or interwoven parts; composite. Involved or intricate, as in structure; complicated. Having parts so interconnected as to make the whole perplexing.” (The American Heritage Dictionary of the English Language, Third Edition). Langefors defined what he called imperceivable systems:

“We define ‘imperceivable system’ to mean a system such that the number of its parts and their interrelations is so high that all its structure cannot be safely perceived or observed at one and the same time” (Langefors, 1973, p. 69)

In addition to these general aspects of complex systems I will refer to a system as ‘complex’ if it displays at least the characteristics of telecommunication systems as described earlier, i.e.,

• frequent interactions between the parts,
• different technologies in the system,
• a constant evolution of the system where new parts must interact with old parts,
• large and globally distributed development projects with many internal deliveries and integration steps,
• many different skills needed,
• actors display many different cultures and traditions,
• turbulent markets,
• frequent organizational changes in the development organization.

Thus I will include in my conception of ‘complex’ also social aspects. In particular I will discuss how society and technology interact in the formation of a shared meaning in a context6.
1.2.5 System

The system concept can be defined as follows: “A group of interacting, interrelated, or interdependent elements forming a complex whole.” (The American Heritage Dictionary of the English Language, Fourth Edition). As such, the system expresses systemic properties which in general cannot be attributed to any of its parts.

I will not engage in a deeper discussion about the definition of systems. However, like Checkland (1981) I want to include a contextual and subjective element 7. In order to be apprehended as a system there must be a viewer who conceives some phenomena in the world as a system. Different system viewers may see different systemic properties. For example, a salesperson trying to persuade a customer to buy a car may bring out other systemic properties of the car than those conceived by the car developers.

1.2.6 Information systems

As noted in the FRISCO report there is no consensus about the essence of what constitutes an information system. An information system can be seen as (Verrijn-Stuart, 2001, p.6):

- A technical system, implemented with computer and telecommunications technology.
- A social system, such as an organization in connection with its information needs.
- A conceptual system (i.e. an abstraction of either of the above).

FRISCO defines an information system in the following way:

“An information system is a sub-system of an organisational system, comprising the conception of how the communication- and information-oriented aspects of an organisation are composed (e.g. of specific communicating, information-providing and/or information-seeking actors, and of specific information-oriented actands) and how these operate, thus describing the (explicit and/or implicit) communication-oriented and information-providing actions and arrangements existing within that organisation.” (Verrijn-Stuart, 2001, p. 72)

Thus both technical and social aspects are included in the definition. In this study I will adhere to this broad view of information systems. In addition to this I will mainly be concerned with information systems used for engineering information management purposes. Such systems are computer systems which handle data related to engineering management activities such as PDM (Product Data Management) systems, ERP (Enterprise Resource Planning) and MRP (Manufacturing Resource Planning) systems.

---

6. For a thorough discussion of complexity in relation to global industrial development projects, see Lilliesköld (2002).
7. This is also the position taken in the FRISCO report in which a foundation for IS development is outlined (Verrijn-Stuart, 2001).
1.2.7 Model

A model can be defined as “A representation of some phenomenon of the real world made in order to facilitate an understanding of its workings. A model is a simplified and generalized version of real events, from which the incidental detail, or ‘noise’ has been removed” (A Dictionary of Geography, Oxford University Press, 1997).

In the Framework a set of models is used (see Section 1.3.1 The Framework). These are used by the actors in the construction of the coordination domain. This means that the models are socially constructed and may not necessarily represent existing phenomena of the real world. Instead the model should be apprehended as composite signs, which signify relevant phenomena in the coordination domain. Signs are discussed in detail in Section 3.2.2 The nature of the sign.

1.2.8 Project

The ‘project’ concept is a concept with many facets (see for example Engwall, 1995). In this study we will understand ‘project’ in the way it is defined at Ericsson: “A project is a named, time-limited and budgeted undertaking for which goals have been established. It is non-recurrent and requires the establishment of a temporary organization.” (ERI-1994-01-01). However, in practice it is problematic to delimit a project precisely, mainly because of the turbulent development situation. Projects are constantly redefined with respect to content, delivery plans and location. One actor at Ericsson expressed this as “The project you finish with is not the same as the one you started with”.

1.2.9 The empirical data

My entire professional life has been within the Ericsson company working with support for the development of complex systems (see Section 1.6 My personal background). Therefore, it was quite natural to choose my empirical data from the Ericsson practice. This is also where the idea of the Framework was born. These circumstances are both an advantage and disadvantage. The biggest advantage is the access to the data; both from my own practice and other areas at Ericsson. Another major advantage is the prospect of seeing my ideas realized and being used in actual development projects, which is the case with the Framework. Thus the relevance of the Framework for Ericsson is well grounded.

A disadvantage is the difficulty in separating my roles as a researcher and as a practitioner. The risk of being biased is obvious. Also, it must be remembered that any achievement in an organization is the result of a collective effort. My contribution to these achievements must be carefully argued for. Furthermore, my results are empirically grounded in the Ericsson practice only, which makes the results harder to generalize.
1.3 The Framework and its grounding theoretical perspective

In this section I will give a short overview of how the Framework is structured and the theoretical perspective the Framework is grounded in. In Chapter 5 the Framework is described in more detail.

1.3.1 The Framework

The concept of ‘framework’ stands for a common, stable and coherent structure, which can be concretized or implemented in different ways\(^8\). Furthermore, this implies that the Framework should be considered as a coherent whole. If one or several of its elements are removed, the Framework as such ceases to exist. As described in Chapter 6 *The history of the Framework*, single elements in the Framework were utilized at Ericsson both before and after the period in which the Framework as a whole existed (roughly between 1997 and 2001).

The Framework consists of three models, an *information system*, a *stabilizing core* and a *strategy*\(^9\) for constructing coordination domains (see Figure 4):

- **The Conceptual Model**\(^10\) signifies the structure of the coordination domain in terms of coordination items and their static relationships. One example of this is that “requirements are allocated to design items”.
- **The Process Model** signifies the dependencies between activities impacting the coordination items, for example that “requirement prioritization” comes before “requirement allocation”. This model corresponds to the definition of coordination given by Malone.

---

8. This complies fairly well with the definition from WordNet ® 1.6, © 1997 Princeton University: a structure supporting or containing something.
9. By strategy, I understand “a long-term comprehensive procedure for achieving a goal or purpose” (Nationalencyklopedin, 2000)
10. The Framework parts are denoted with initial capital letters all through the dissertation.
• The Transition Model signifies how the coordination activity domain interacts with other activity domains, that is how coordination items are interpreted and translated when they appear in other contexts. For example, out of all the information produced in a software design context only some is important in the coordination domain, such as when status or revision of a source code file has changed.

• The models are implemented in the Information System (IS) and used in the usage domain.

• The Stabilizing Core consists of rules, norms, standards, etc. which provide the necessary stability in the domain.

• In order to achieve a shared meaning about the coordination domain a strategy is included in the Framework, the Domain Construction Strategy. This strategy consists of an ongoing iteration between reflection and action phases in which the models and the IS are alternately reflected upon and tried out in practice. In this process the models and the IS are conceived as signs which mediate the shared meaning among actors. Furthermore, the iteration between development in the coordination domain and usage in the usage domain means that the implementation in the IS is continuously changed. Thus, an evolutionary information system development approach is utilized.

The particular syntax and semantics of each model as well as what particular IS to use, are not prescribed in the Framework. However, during the validation of the Framework in the Ericsson practice particular realizations of these were used. The result of applying the Domain Construction Strategy is a gradual construction of the elements in the Framework including a shared meaning among the actors in the coordination context.

1.3.2 The Activity Domain Theory

The Framework as described in the previous section is grounded in a specific theoretical perspective which explains why the Framework is structured as it is. Since both the artefacts and the shared meaning are constructed, this implies that the Framework is based on a social constructivist ontology. This means, that in order to define the Framework, we need a theoretical perspective which is rich enough to include both individual, social and technological aspects of coordination. In Chapter 3 The Activity Domain Theory a theory for human activity is proposed. This theory contains the following elements, called coordination constituents of human activity:

• Intersubjectivity: This is a prerequisite for a concerted action in a context. Shared meaning arises in the interaction between humans and is mediated by various sign systems, above all language.

• Contextuality: Humans have an inherent capability to apprehend contexts and change focus between different contexts. When the focus is changed, spatial and temporal structures are reorganized in the sense that new phenomena and action patterns become attended to while other diminish below the attention horizon. The Conceptual

11. The information system used is eMatrix from Matrix-One Inc. This system was earlier called just ‘Matrix’. I will use ‘eMatrix’ throughout the dissertation.
INTRODUCTION

Model in the Framework signifies the context of the coordination domain to the actors in this domain.

- **Domain transition**: The practice of humans takes place in an unreflected, common sense understanding of the world where phenomena are taken for granted. In order to understand the essence of the phenomena these must be revealed, reinterpreted and translated in order to be comprehensible. In order to do so a transition must take place between the two domains of discourse. The Transitional Model in the Framework signifies the transitions between domains.

- **Experiential learning**: Human knowledge and capabilities are acquired in an ongoing iteration between reflection and action. The Domain Construction Strategy in the Framework is grounded in the experiential learning constituent.

- **Orientation**: Humans have an inherent capability to structure the world spatially in a context. The orientation is achieved by classifying and categorizing phenomena in a context and how they are related to each other. The spatial structure is static in the sense that phenomena and their relations change only slowly as experiences are accumulated in the individual. The Conceptual Model in the Framework also signifies orientation to the actors in the coordination domain.

- **Temporality**: Humans have an inherent capability to structure the world temporally which relates to the order in which events occur in a context and how the individual reacts to and impacts these events. The temporal structure is interrelated to the spatial structure. The Process Model in the Framework signifies the temporality constituent.

- **Stabilizing core**: Stabilizing structures are necessary in human activity. A stabilizing core is a prerequisite for actions, and provides a proper balance between order and disorder. The Stabilizing Core in the Framework is grounded in the stabilizing core constituent.

- **Tool usage**: The making and use of tools are inherent in human activity. The Information System in the Framework is grounded in the tool usage constituent.

- **Motive, outcome and object**: Human activity has a motive which is the reason why the activity exists. In the activity actors work on object(s) in order to produce an outcome.

The idea behind labelling these elements ‘coordination constituents’ is that I claim that they elicit fundamental coordinating elements in human activity. The constituents are grounded in praxis philosophy (Kosik, 1976), cognitive sciences (Gärdenfors, 2000a), semiotic sciences (Voloshinov, 1986/1929) and Activity Theory (Engeström, 1999; Kuutti, 1991). From the constituents the construct of Activity Domains is defined, which may be apprehended as praxis concretized for explanatory and constructive purposes.

### 1.4 Expected research results

In this section I will discuss the expected results from the study. From the main research question: “What are the impacts on coordination from the Framework intervention?” detailed research questions may be formulated. I will also characterize what type of knowledge they provide. This is needed in order to decide on strategies and methods for investigating them. The methodological issues are discussed in Chapter 2 Research design and realization.
1.4.1 Detailed research questions

The detailed research questions are formulated as follows:

**RQ1: How did the Framework evolve in the Ericsson development practice?**
This research question provides knowledge of the introduction, diffusion and absorption of the Framework in the Ericsson practice. The Framework history stretches over more than a decade and has encountered several obstacles, both technical and social in nature. The expected knowledge contribution of this question is of the *reconstructive* type\(^\text{12}\). This type of knowledge reconstructs a series of events (“what happened here?”).

**RQ2: What are the overall consequences from the Framework intervention in the Ericsson development practice?**
This research question provides knowledge of the overall consequences of the Framework during its trajectory in the Ericsson practice. The expected knowledge contributions from this research question are of the *explanatory* type. The Framework intervention (cause) brings of certain consequences.

**RQ3: Which elements in the Ericsson development practice contribute to successful outcomes of development projects?**
This research question provides knowledge of such elements in the Ericsson development practice which experienced actors consider important to successful outcomes of development projects\(^\text{13}\). The expected knowledge contribution from this research question is of the *explanatory* type (“what caused these effects?”) since it provides knowledge of what elements (cause) lead to successful development outcomes (effects). In this study, these elements are called *reinforcement rods* (see Chapter 8).

**RQ4: What are the impacts on coordination from the Framework intervention in the Ericsson development practice?**
This research question provides an answer to the main research question. The expected knowledge contribution of this research question is of several types. First, it is of the *explanatory* type since it explains the consequences the Framework (cause) has on coordination. Furthermore, this knowledge is *transformational* (“how shall we change this?”), since the intervention of the Framework changes the Ericsson coordination practice. Finally, this knowledge is of the *innovative* type (“this is possible”), since the Framework opens up possibilities previously not imagined in the Ericsson practice. One example of this is the possibility of replacing all current ISs for coordination with the one in the Framework.

\(^{12}\) The classification of knowledge types comes from Goldkuhl (1998, my translation).
\(^{13}\) For a thorough discussion of success and failure factors in relation to global industrial development projects, see Jonsson (2002).
INTRODUCTION

In the table below the research questions are summarized.

Table 1. Research questions

<table>
<thead>
<tr>
<th>Detailed research question</th>
<th>Knowledge contribution</th>
<th>Knowledge type</th>
</tr>
</thead>
<tbody>
<tr>
<td>RQ2: What are the overall consequences from the Framework intervention in the Ericsson development practice?</td>
<td>Identified consequences</td>
<td>Explanatory</td>
</tr>
<tr>
<td>RQ3: Which elements in the Ericsson development practice contribute to successful outcomes of development projects?</td>
<td>Identified elements</td>
<td>Explanatory</td>
</tr>
<tr>
<td>RQ4: What are the impacts on coordination from the Framework intervention in the Ericsson development practice?</td>
<td>Identified impacts on coordination</td>
<td>Explanatory Transformational Innovative</td>
</tr>
</tbody>
</table>

In Figure 5 the derivation of the knowledge contribution is illustrated. The knowledge contributions from RQ1 (the history of the Framework) and RQ2 (the overall consequences of the Framework) follow from the analysis of the Framework intervention in the Ericsson practice. In Chapter 4 Articulating Coordination, I propose a conception of coordination which is an elaboration of the definition of Malone & Crowston (1994). Besides this grounding in the literature, the conception is also grounded in data from the Ericsson practice. The knowledge contribution from RQ3 (elements contributing to successful outcomes of development projects) follows from RQ1, RQ2 and an analysis of projects in the Ericsson practice where the Framework was not used. From the conception of coordination and the knowledge contributions from RQ1, RQ2 and RQ3 follow the knowledge contribution of research question RQ4.

1.4.2 Additional knowledge contributions

In addition to the contributions from the research questions, the following contributions can be expected:
• The Framework regarded as an artefact can be considered a knowledge contribution in the sense that it is the outcome of a human activity with the motive of providing coordination.

• The Activity Domain Theory outlined in Chapter 3 may be considered as an alternative theoretical point of departure for IS development where both social and technical aspects are considered. An evaluation of some other theories is also given, see Section 3.5 Alternative perspectives on human activity, p. 85.

• The conception of coordination as an activity domain given in Chapter 4 Articulating Coordination may be considered as a contribution to the literature concerning coordination.

These additional knowledge contributions should be regarded as spin-off contributions resulting from the research questions in the study. There are no research questions associated with these contributions and the results are preliminary at most.

1.4.3 Characterizing the knowledge contributions

What constitutes the originality of the knowledge contributions in this dissertation? First of all, concrete results have been achieved in an application area of immense size and complexity. Between 1999 and 2002 approximately 140 main projects and sub-projects at Ericsson have used various parts of the Framework (ERI-2002-06-06). These projects were distributed over more than 20 different development units around the world and carried out in a fiercely turbulent environment. The results from applying the Framework indicate that some systems could not have been developed without the approach suggested in this dissertation (see Chapter 7 Framework consequences).

The Framework consists of elements that are well-known taken one by one. However, I claim that the juxtaposition of these elements according to the Activity Domain Theory is new. Also, the inclusion of sign mediated shared meaning in the Framework makes it possible to consider both individual, social and technological issues, something that has been on the information system research agenda for a long time (see for example Iivari & Lyytinen, 1998). However, the main results appear to be analytical. Few operational or constructive results have been reported. I claim that the Activity Domain Theory is indeed capable of leading to constructive results in the form of the Framework, which has proven to be operational in the Ericsson practice.

1.5 Related work

References to the literature will mainly be given in each chapter. Here, we want to discuss some references which are related to the overall purpose of the study. Grinter (1999) has made a study of how systems architects cooperate at a telecommunication equipment vendor. Eriksson et al. (2002) have made a case study at the Asea Brown Bovery (ABB) company. Both these studies were carried out at organizations which in many respects face similar challenges as Ericsson.
Grinter’s findings indicate that many systems fail because no-one cared about the total structure of the system. This has incited the role of the system architects, the purpose of which is to maintain the coordination across organizational units and negotiating conflicting interests.

Eriksson et al. (ibid.) states that the execution of a globally distributed high technology project is a great challenge. The cultural issues must be handled since several organizations may be involved, each having a culture of its own. They identify a number of prerequisites for improving such projects:

- Frequent project meetings in order to create a mutual trust and a common language. One tool for achieving this is a *dependency diagram* which shows the dependencies in the project.
- A globally accessible database which makes it possible to exchange information efficiently.
- The overall project management should concentrate on the project control process and leave the local projects to work according to well known project models and traditions.
- The status of the project should be presented in a compact and efficient manner. An example is the use of a traffic light metaphor: green for deliveries on time, yellow for deliveries with a high risk of being late and red for delayed deliveries.
- Flexibility is more important than detailed definitions of responsibilities.
- The project organization should be set up in such a way that each local project develops a number of products which are as independent as possible of other products.
- The key success factor is the dependency diagram. It is a very simple and powerful tool for anchoring the goals of the project in the local organizations and local projects.

These issues are also found in the Ericsson practice and most of them are addressed by the Framework.

A framework for modelling and analysing Engineering Information Management systems in product development and manufacturing organizations is proposed by Svensson et al. (1999). The framework consists of four interrelated models for process, information, role and computer systems aspects. The relationships between these models are visualized by Design Structure Matrices (Eppinger et al., 1994). The main purpose of the framework is to perform a thorough analysis of an existing state in order to lay a solid ground for future developments. The authors acknowledge the inherent complexity and the difficulties in arriving at a shared meaning. They also state that in order to master the complexity, several views are necessary.

In comparison with the Framework proposed in this dissertation, the framework of Svensson et al. (ibid.) is mainly focused on technical aspects. It lacks a theoretical perspective of human activity which might guide the selection of elements to be included in the framework. For example, the domain transition and stabilizing constituents are not treated. Moreover, the constructive aspect is not included, that is, there is no guidance in their framework of how an existing situation should be improved or how a common understanding can be achieved.
CHAPTER 1

The notion of ‘coordination’ is extensively treated in the literature (see e.g., Larsson (1990) or Malone & Crowston (1994) for a comprehensive survey). A recent contribution is given by Melin (2002) who analyses coordination within organizations and between organizations in intra-organizational relations.

Moreover, coordination is closely related to the technical aspect of ‘management’. In this sense there exists a large body of knowledge concerning the management of the development of complex systems. For example, Reinertsen (1997) emphasizes the economic aspects. Oliver et al. (1997) focus on the technical aspects, whereas the human aspects of management are in the background. Some contributions are domain specific, such as Royce (1970) who treats the management of large software development projects. Another group of contributions concerns how to structure complex design projects into manageable tasks by focusing on the technical dependencies in the system (see for example Eppinger et al., 1994). Suh (1990) suggests a strategy called Axiomatic Design which aims at reducing the complexity of the design and thus the need for management. Sage (1995) provides a thorough analysis of system engineering and system management.

However, in most of these contributions human activity is treated in a superficial way. Moreover, they are in general not grounded in a theoretical perspective on human activity. Thus the social aspects of coordination are often subdued in favour of the technical ones.

In contrast, the human activity perspective has been prominent in some traditions in IS development research for many years, especially in the Scandinavian IS development approach (Iivari & Lyytinen, 1998). One of the early pioneers was Ehn who argues that a thorough understanding of information systems can be achieved only in practice:

“My point is that computer science and systems design has to a large degree been unsuccessful in relating design knowledge as detached reflection to design knowledge as practical skill. The latter has been made invisible.” (Ehn, 1988, p.41)

Ehn also points out that the system design activity should be an instrument to promote a common understanding among designers and users, and that the design process should be carried out in the specific historical and social setting in which the artefact is used.

The philosophical perspective of Ehn is similar to the one in the Activity Domain Theory. Both are grounded in the Marxian praxis tradition. However, Ehn has an emancipatory ambition which is not present in the Activity Domain Theory. This theory is closely related to, albeit structured differently than the Activity Theory (see for example Engeström, 1999 or Kuutti, 1991). Activity Theory also sees the totality of human activity as the proper analysis unit. However, it has been used mainly for analytical purposes (Persson, 2000, Virkkunen & Kuutti, 2000).

Persson (ibid.) describes IS design for autonomy and control in military command work. Most constituents in the Activity Domain Theory can be found in his work. For example, the core design challenge is how to reconcile demands for power, autonomy and control with demands for stability. Persson states that an organization which can balance these aspects becomes a viable system, that is, it has the freedom for action. This issue is directly related to the stabilizing core constituent in the Activity Domain Theory.
INTRODUCTION

However, the bulk of the works in the literature seems to be analytical in nature. The combination of coordination of complex system development tasks, social and technological aspects of human activity and practical application in industrial settings appears to be rare.

1.6 My personal background

I have been working with support systems for developing complex telecommunication systems for more than 30 years. This has all been done within the Ericsson company, which is a large telecommunications’ system provider worldwide. Areas where I have been working include:

- Filter design for frequency multiplexed transmission systems (1970s).
- The stochastic optimization of transmission systems (1970s-1980s, see for example Kjellström & Taxén, 1981).
- Designing support systems for evaluating fibre optical transmission systems. This included writing software in FORTRAN, PASCAL and Ada (1980s).
- Process development for hardware- and software design (1990s).
- Method and tool support development for incremental development (1990s).
- Support for coordinating the development of telecommunication systems (late 1990s).

In doing so, I have had various positions: designer, method developer, line manager, project manager, department specialist, etc. On some occasions I have been participating in large development projects involving several thousand persons worldwide. On most occasions, these projects have been successful. However, some have been spectacular failures. One conspicuous phenomenon in these projects was the inability to comprehend dependencies. There were literally thousands of schemes, models, descriptions, etc. that described the system itself. The dependencies between different subprojects were unclear. The dependencies between organizations and responsibilities were unclear. In short there was no comprehensible view that could be used to coordinate the activities in the project. The result was management chaos. No one could tell the exact status of the project. Were the deliveries on time? What parts did they consist of? How did the dependencies between them look?

Out of these experiences the need came to somehow find a better way of coordinating large system development projects. The big issue of course was how to do this. In the middle of the 1990s I became involved in a task of developing an incremental development method package. This package would replace the traditional waterfall model for software development, which had been in use for a long time at Ericsson. It was becoming painstakingly clear that the waterfall method had severe disadvantages in the ever increasing turbulent environments we were facing.

The method package was an attempt to consolidate the experiences of development projects which had tried some variant of incremental development. Thus, the task was in essence to arrive at some shared meaning of what constituted a good incremental development method. However, this turned out to be very hard to achieve. After a long period of endless discussions without getting anywhere, we started to use conceptual models as a discussion tool, just to have something concrete to point at instead of just talking. It was still as hard to agree but at least the discussions began to converge. After about six months we had the first ver-
sion of a method package ready. So, out of this experience the need came to somehow find a way to facilitate the evolution of a shared meaning among a group of actors.

Another source of the study was of a more philosophical nature. I often felt that many concepts used were unclear. For example, the often used concept of a “function” is in fact not clearly defined at all. A function is almost always considered a property of a product or system. However, that is not the way we use this concept in daily life. The function of a torch, for example, can be to light a room, warm somebody, set fire to a building, be a symbol for the Olympic games, etc. Thus, the essence of a function is dependent on its context of use rather than being an inherent property of an object. So, my experience from this is that the fundamental ideas are very important since they direct our thinking and set delimitations for actions.

Out of these and many other experiences, the idea grew of a framework to support the coordination of complex systems’ development. Somehow, I imagined, it would be possible to bring together important coordination aspects into a coherent framework. From the very first ideas to the current implementations I have been the promotor and ‘entrepreneur’ in introducing the Framework in the Ericsson practice, often in conflict with other interests. It is the result of these hunches that are reported in this study.

1.7 Delimitations

The following delimitations apply to the study:

- **Management of people.** Coordination and ‘management’ are kin concepts, which is illustrated in the definition by Malone & Crowston (1994): “coordination is the management of dependencies between activities”. However, ‘management’ may also include leadership issues like motivating actors, solving conflicts, team-building, etc. These aspects are not included in the study.
- **Study field:** My empirical findings have been collected from the Ericsson company only.
- **Experiments:** There are no controlled experiments, simply because it is difficult to perform them within the study field.
- **Complex systems:** There are many complex system besides the ones treated in the study, e.g. biological and economic ones. These are not included in the study.
- **Elements contributing to successful outcomes of development tasks:** These elements are grounded in the Ericsson practice, mainly by interviewing experienced actors. In the literature, there are numerous results reported concerning success factors for development tasks (see for example Jonsson, 2002). I will not include a discussion about how the elements found in the Ericsson practice relate to those reported in the literature. Furthermore, I will concentrate on those
elements which are closely related to coordination (see Figure 6).

• The product life-cycle: A product such as a network for the 3rd generation of mobile systems is subject to many activities during its life time. These include marketing and selling the product, developing the product, producing it and providing maintenance and services of installed products. In the study my focus is product development. However, I claim that some results are valid for the whole life cycle, for example the notion of activity domains.

In addition to these delimitations there is of course the limited time available for the thesis work. It is simply impossible to cover everything. The approach of focusing on human activity as the unit of analysis opens up vast areas of relevant scientific research, such as the Computer Supported Cooperative Work (CSCW) field, the Human Computer Interface (HCI) community, ethnography, ‘communities of practice’ research, distributed cognition and cognitive sciences to mention but a few. I am quite aware of the possibility that there are research results within these areas which are relevant for my work that I may have overlooked.

1.8 Stakeholders
There are various stakeholders who might be interested in this study:

Actors at Ericsson
These are actors directly involved in the development practice: project managers, configuration managers, test configuration managers, requirement coordinators, etc. Their interest is to use the Framework to improve their practice.

Other actors are the top management of the company since the findings may impact the finances of the company. For example, if the coordination of large and costly projects is indeed improved, this may reduce the cost of these projects. Furthermore, there exist a large number of information systems at Ericsson which are used for coordination. The results indicate that many of these information systems can be replaced by the one in the Framework (see Section 7.4 Information System Architectures).

The scientific community
This includes my supervisors, other colleagues at the university and myself: we are of course interested in the knowledge produced and that it fulfils the quality norms of a Ph.D. dissertation. Furthermore, this includes lecturers at the university: persons who have held courses and seminars which have contributed to the knowledge content in the study.

Research colleagues in product development, organizational learning, computer science (large, industrial, software development methods), knowledge management, IS sciences and the like may find interesting results in the study.

Interviewees
A special group is my interviewees: persons that I have interviewed, both within and outside Ericsson.
CHAPTER 1

Other organizations
To the extent my findings are generalizable, other organizations developing complex systems should be interested. This includes other companies developing telecommunication systems, the car industry, the air industry, the electro-mechanical industry and the like.

Financing parties
This includes all parties that have contributed to financing my study:

- The Swedish National Board for Industrial and Technical Development, NUTEK, project P10518-1.
- The Industry Research School in Applied Information Technology and Software Engineering at Linköping University, funded by the Foundation for Knowledge and Competence Development.
- Ericsson AB.

Vendors of information systems
The IS used in the Framework belongs to a class of information systems called Product Data Management (PDM) systems. These complex systems represent huge commercial interests and are often very complicated to implement and deploy globally in large organizations. The evolutionary strategy utilized in the Framework requires a flexible and easy-to-modify PDM system. If my results show that this kind of strategy results in improved implementations, it might be of interest to the vendors of PDM systems to provide systems that comply with the needs of the Framework.
1.9 Structure of the dissertation
The dissertation consists of four parts. Part I *INTRODUCTION and RESEARCH DESIGN* contains the introduction of the dissertation (this chapter). In Chapter 2 *Research design and realization* I discuss some scientific positions taken and how the research has been carried out.

In Part II *THEORY and DOMAIN CONSTRUCTION* I first propose a theory which has governed the design of the Framework (Chapter 3 *The Activity Domain Theory*). A main concern has been to find a theoretical perspective which includes both individual, social and technological issues. The reason for this is my conviction that the problems of coordinating the development of complex systems of the kind we are concerned with in this study cannot be mastered without including these issues. The Activity Domain Theory is grounded in praxis philosophy, cognitive sciences and philosophy of language. I also make a brief comparison with alternative theoretical approaches.

Next, I define my conception of coordination in Chapter 4 *Articulating Coordination*. The conception is an elaboration of the definition of coordination given by Malone & Crowston (1994) in which influences from the Ericsson practice have been included. In short, I understand coordination as a particular kind of activity where actors work together to provide coordination in the development projects.

Chapter 5 *The structure of the Framework* contains a detailed description of the structure and elements of the Framework. I describe the rationale behind the three models, the IS and the experiential domain construction strategy.

Part III *RESULTS FROM THE ERICSSON PRACTICE* starts with a reconstruction of the history of the Framework in the Ericsson practice (Chapter 6 *The history of the Framework*). An analysis of the history is also included. This provides an answer to research question number 1.

Next, in Chapter 7 *Framework consequences* I report the overall consequences of introducing the Framework in the Ericsson development practice. This provides an answer to the research question number 2. This chapter also contains a discussion of different information system architectures at Ericsson.
In the following chapter, Chapter 8 Reinforcement rods, elements called reinforcement rods are identified, which experienced actors at Ericsson consider important to successful outcomes of projects. The data for this research question have been collected both from projects where the Framework was not used and from projects where it was used. This provides an answer to research question number 3. In addition, the relationship between the reinforcement rods and the Framework is analysed.

In part IV DISCUSSION & CONCLUSIONS the results are discussed. In Chapter 9 Discussion of the Framework approach the Framework and its elements are discussed. This is followed by a discussion of the coordination of development tasks. Furthermore, the Activity Domain Theory is discussed as well how the results can be generalized. Altogether, this provides an answer to the research question number 4.

Finally, Chapter 10 Conclusions contains the conclusions which can be drawn from the study. Some suggestions for further research are given. The chapter also contains an evaluation of the research design. It is concluded with some reflections on the significance of the study for Ericsson and my role as a researcher and actor in the Ericsson development practice.

1.9.1 Reading guidelines

In the table below I have indicated my subjective opinion about the chapters of most interest for some stakeholders:

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Actors at Ericsson</th>
<th>The scientific community</th>
<th>Other companies</th>
<th>Financing parties</th>
<th>PDM vendors</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Introduction</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 Research design and realization</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 The Activity Domain Theory</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 Articulating coordination</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 The Framework</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6 The history of the Framework</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7 Consequences of the Framework</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8 Reinforcement rods</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9 Discussion of the Framework approach</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10 Conclusions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Chapter 3 is a demanding theoretical chapter and requires a fair amount of familiarity and interest in philosophy, cognitive sciences and semiotics. The main purpose of the chapter is to give a rationale why the Framework is structured the way it is, and it is not necessary to read this chapter to appreciate the rest of the dissertation. The results in part III (Chapters 6 - 8) should be interesting reading for practitioners working on coordination of large and complex development tasks, both within and outside the Ericsson company. For PDM Vendors, the indication points to chapters where information system issues are discussed.
2 Research design and realization

Some general quality requirements on a dissertation are that its knowledge contributions are original, communicable, trustworthy and relevant. The purpose of the research design is to secure the trustworthiness in terms of credibility, validity, controllability and reliability.

2.1 Positions taken

In this section I discuss the positions I have taken on some issues concerning the research design. In the next section, 2.2 Realization, I describe how the research design was carried out. These sections are interconnected in the sense that the positions taken have guided the choice of strategies and methods used.

2.1.1 An interpretative research tradition founded in praxis

The study concerns the coordination of the development of complex systems in the Ericsson company. The way coordination is apprehended is largely a matter of agreements made by the actors in the company. Certain facts hold true because there is a shared understanding about the meaning of key concepts. Thus the study object includes interpretations made by humans:

“The objects of thought, which researchers in social science construct, refer to, and are grounded in, the objects of thought already constructed in ordinary thinking of people living their lives together with others in this everyday world.” (Schütz, 2002/1962, p. 20, my translation)

The theoretical point of departure in praxis philosophy in the study is also in line with this (see Chapter 3 The Activity Domain Theory). The ontology and epistemology of praxis can be summarized as follows: There is an external world outside human beings which is distinct from, but related to, humans. Individuals acquire their knowledge and capabilities by interacting with this world. The interaction is mediated by signs1. Human action is socially oriented which means that individual understanding is grounded in the social, intersubjective conditions of mankind, above all a language. In essence this means that the world of coordination is a socially constructed one based on an independent, but to humans related,

---

1. For a comprehensive discussion of signs and semiotics, see Innis, 1985.
reality. The praxis perspective can roughly be mapped onto the classification of Walsham (1995) in the following way:

<table>
<thead>
<tr>
<th>Epistemology</th>
<th>Ontology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positivism: Facts and values are distinct and scientific knowledge consists only of facts</td>
<td>External realism / objectivism: Reality exists independently of our construction of it.</td>
</tr>
<tr>
<td>Non-positivism: Facts and values are intertwined. Both are involved in scientific knowledge</td>
<td>Internal realism: Reality-for-us is an inter-subjective construction of the shared human cognitive apparatus</td>
</tr>
<tr>
<td>Normativism: Scientific knowledge is ideological and inevitably conducive to particular sets of social ends</td>
<td>Subjective idealism / relativism: Each person constructs his or her own reality</td>
</tr>
</tbody>
</table>

The purely positivistic epistemology and subjective idealism / relativism ontologies are not found in the praxis perspective.

The research questions in the study, the expected knowledge contributions and knowledge types are described in chapter 1 Introduction. This is repeated here for convenience:

<table>
<thead>
<tr>
<th>Detailed research question</th>
<th>Knowledge contribution</th>
<th>Knowledge type</th>
</tr>
</thead>
<tbody>
<tr>
<td>RQ2: What are the overall consequences from the Framework intervention in the Ericsson development practice?</td>
<td>Identified consequences</td>
<td>Explanatory</td>
</tr>
<tr>
<td>RQ3: Which elements in the Ericsson development practice contribute to successful outcomes of development projects?</td>
<td>Identified elements</td>
<td>Explanatory</td>
</tr>
<tr>
<td>RQ4: What are the impacts on coordination from the Framework intervention in the Ericsson development practice?</td>
<td>Identified impacts on coordination</td>
<td>Explanatory</td>
</tr>
</tbody>
</table>

The setting of the research in one company, the nature of research questions and the praxis perspective implies that our study can be classified as an interpretative one (Walsham, 1995) in which we will use research methods from qualitative research approaches (Yin, 1989), mainly the case study. It is possible to develop some quantitative knowledge concerning the effects of the Framework, but mainly understanding, meaning, interpretation and human action will be in focus.

2.1.2 The roles of theories

The purpose of this section is to declare the usage of theories in the study. We will use the definition given by Strauss & Corbin (1998):

"[A theory is a] set of well-developed concepts related through statements of relationship, which together constitute an integrated framework that can be used to explain or predict phenomena."

(Strauss & Corbin, 1998, p. 15)
At least the following roles of theories can be identified in the literature (Yin, 1989; Walsham, 1995; Alvesson & Sköldberg, 1994):

- **Theory as an initial guide for data collection**: When faced with a complex and ill-structured research field there is no guarantee that the data collected will be valid or of interest, no matter how long the researcher stays in the field. The theory may ‘sensitize’ the researcher to a particular set of concepts which might otherwise have been ignored. The theory functions as a screening device or searchlight which filters out irrelevant phenomena.

- **Theory as a guide for data analysis**: This use of theory is similar to the previous one with the difference that the theory is used to analyse a given set of data. For example, a rich and ‘thick’ empirical collection of data in the form of interviews may be analysed from various perspectives using different theories as sensitizing devices.

- **Theory as a final result**: The theory may be the outcome of the research in a theory-generating process. This is the case in explorative studies where the research interest is to “find out what is going on here”.

- **Theory as emergent in an iterative process**: The research process may start with an initial formulation of a theory which is used to collect an initial set of data. This data may in turn refine the theory which is again used to collect another set of data.

- **Theory informing constructive and / or operational activities**: In this role the theory provides normative knowledge for practitioners trying to achieve operational or constructive goals.

In this study we take the position that the research process consists of an interplay between theory and empirical findings. The relative balance between theory and empirical findings may vary during the study. Depending on the state of the theory and the findings we may talk about the permeation of theory and data in the study. At the extremes a high theory permeation indicates that a theory exists and remains unaffected during the study. Likewise, a high data permeation indicates that a rich set of data exists and remains stable during the study.

In a deductive approach the theory permeation of the study is high and the purpose of the research process is to proceed from a low data permeation to high in order to validate the theory. In the inductive approach the data permeation is high and the purpose of the research process is to proceed from a low theory permeation to high in a theory generating process. The abductive approach moves from a low data and theory permeation towards a high permeation in both theory and data by oscillating between theory validation and theory generation. Thus, abduction can be seen as encompassing several cycles of oscillation between deduction and induction (Alvesson & Sköldberg, 1994). These different aspects of theory can be illustrated as in Figure 8:
2.1.3 A single-case, multiple-units of analysis study

A case study is an investigation of a particular, delimited segment of society which can be associated with individuals, groups, organizations, institutions, projects, etc. (Yin, 1989). Case studies are applicable in situations where describing, understanding, and explaining are of major concern, i.e. in interpretative research traditions. Furthermore, the case study may be the only viable alternative in situations where a random selection of samples cannot be performed (ibid., 1989).

This is applicable to this study. It is confined to one company only: Ericsson. Ericsson is a large, trans-national company with around 60,000 employees around the world. These actors work with different tasks related to many different products.

All the data have been collected from Ericsson, and Ericsson is the only company I have worked with during my professional life (see Section 1.6 My personal background for more details). This also means that I have had full access to the data, both as documents and other artefacts like the information system in the Framework. Also, I have had no problems in securing the cooperation of informants.

On an overall level, Ericsson can be seen as comprised of four cooperating activity domains:

- **Research and Development**: The purpose of this activity domain is to develop products at development units worldwide. Each product is sold on many markets, produced by many supply units and serviced by many service units.
- **Markets and Sales**: The purpose of this activity domain is to market and sell total tele-
communication solutions to customers at market units worldwide. There are many markets, and each market unit works with many products and many supply and service units.

• **Supply and Implementation:** The purpose of this activity domain is to produce and install total solutions at customer sites. This is done by Flow Control Centres worldwide. Each centre services many products and markets.

• **In Service Support:** The purpose of this activity domain is to service and upgrade installed solutions at customer sites. This is done by service units worldwide. Each service unit supports many products and is active on many markets.

The data have mainly been collected from the Research and Development activity domain. The delimitation of the case in time is more complicated. The evolution of the Framework started long before it became an object of scientific research in 1998. This means that there are two perspectives involved. From the inception of the Framework in the early 1990s until 1998, the perspective was only to provide the Ericsson development with an improved coordination support. After 1998, the scientific perspective was added while the practical perspective still remained. I place the end point in time for the study at the second half of 2002 when I started to write up my dissertation.

**Unit of Analysis**

A major concern in any case study is the choice of the Unit of Analysis (UoA):

"The key issue in selecting and making decisions about the appropriate unit of analysis is to decide what it is you want to be able to say something about at the end of the study." (Patton, 1990, p. 168)

In this study, three major UoAs can be identified: the Framework, project and activity domain. The Framework and its elements are in focus for research questions 1, 2 and 4. Concerning the third research question “Which elements in the Ericsson development practice contribute to successful outcomes of development projects?” the natural UoA is the project. Aspects of projects include among other things lead time, the number of actors, size, geographical distribution, classification (new, traditional), complexity, etc.

The last UoA is the activity domain as described in Section 3.4 *The Activity Domain Theory – structured praxis*, p. 84. I take the position that any UoA regarding human activity must encompass the totality perspective inherent in for example the praxis perspective. This is a similar approach to the one taken in Virkkunen & Kuutti (2000). They argue vigorously for the importance of choosing the correct UoA. In spite of an overwhelming stock of research in organizational learning there is a striking scarcity of practical guidelines to increase organizational learning. The authors claim that the major reason for this is the absence of a proper UoA for organizational learning. Instead of focusing on ‘organization’, ‘individual’, ‘team’ and the like, they propose the ‘Activity System’ as a proper UoA. The Activity System “… makes it possible to analyse the specific historical, local challenges and problems of organizational learning and to direct a collective learning process” (ibid., p. 291).

The choice of the activity domain as the UoA impacts how the information system (IS) in the Framework is viewed. In general, in the IS discipline the IS is quite naturally taken as the UoA. For example, Dahlbom & Mathiassen take the computer system as the UoA: “The
computer is on stage” (Dahlbom & Mathiassen, 1993, p. 1). Even though they put the computer / IS in its social context and propose a dialectical approach towards IS development, the IS is in focus and the human activity in the background.

However, when the activity domain is taken as the UoA, the IS in the Framework is regarded as a tool which interacts in a dialectical way with the other constituents of the activity domain. Thus, the activity domain is in focus and the IS in the background. This means for example that if the activity domain is changing and evolving, then the IS has to evolve too. An evolutionary approach towards IS development becomes the natural approach which in turn puts harsh requirements on the properties of the IS. For example, it must be very easy to modify its implementation in the activity domain.

In summary, the study can be classified as longitudinal single-case, multiple-unit of analysis study. Thus it is of the type 2 in the classification of Yin (1989).

2.1.4 Action research

Action research has been characterized as “[the] continuous interaction of theory and practice.” (Baskerville & Wood-Harper, 1996, p. 240). At Ericsson I have been promoting the Framework from the early 1990s until now (2002). My scientific studies started in 1998. This means that the practical aspect has been present all throughout this period. The theoretical aspect of praxis was guiding this practice, however in a vague and unarticulated way. This perspective was articulated during the research period when a conscious interaction between theory and practice took place. Thus my role may be termed ‘action’ until 1998 and ‘action research’ from that point on.

The action research role implies that special attention must be paid to issues concerning the credibility of the results. Credibility may be apprehended as an assurance that there exists empirical evidence for results claimed and that reasonable interpretations have been made (Svensson, 1996). I have handled this as follows:

- **My contributions:** I claim as my contributions only such results that can be singled out as clearly original or identifiable as my own. This is grounded in the data (documents, interviews, etc.) or in my own scientific contributions (workshops, conferences, seminars, etc.). Whenever I have been participating in a collaborative effort I describe my role in this effort. Basically this means that I try to avoid exaggerating my role and only include such results that can be acknowledged by other sources.

- **The separation of roles:** Since I have been working both as an actor in the Ericsson context and as a researcher in the academic context, I have tried to separate these roles as clearly as possible. The reason for this is that the concepts, expressions, ways of thinking, etc. in the academic context are alien to the Ericsson context. These two contexts have to be ‘bracketed’ in order not to cause confusion. As the link between these contexts I have displayed aspects of either context in the other, but tried to make these aspects as lean as possible. My team members at Ericsson have always been informed that I am doing research, but they have only noticed this in the form of results from the research, for example, the information system in the Framework. The underlying ideas, intentions, etc. in my research have not been discussed at Ericsson, simply...
RESEARCH DESIGN AND REALIZATION

because the interest from the organization has been very modest.

• Report plus and minus: I have tried to account for both positive and negative results in the study.

• Bias: The Framework is by and large an initiative from the “floor”, often in direct conflict with other approaches sanctioned from top management. Furthermore, I have had a strong personal interest to see the Framework implemented and used. One consequence of this was that I had to do intensive lobbying and form alliances with powerful actors who shared my belief (among all project managers). Naturally, there are also personal motives involved like recognition for my effort, etc. This situation creates problems with bias. The measures I have taken to cope with bias are first of all to declare my own perspective. This is done in Section 1.6 My personal background. Secondly, I have tried to account for other opinions and interpretations.

• Loyalty: A particular form of bias springs from my long employment with Ericsson (more than 30 years). Due to loyalty with the organization I might be inclined to suppress results that are negative for Ericsson and exaggerate positive results. However, Ericsson has not exercised any pressure to adjust my results (except for excluding company sensitive information regarding financial and project lead-time data). On the contrary, an open (bordering on uninterested) attitude towards my results has been prevalent all throughout my research. I have striven to be aware of the risk of being loyal with the company when reporting my results. This risk was considerably reduced when I was dismissed from Ericsson in 2002.

• Interpretations: Since the research object includes human actions and interpretations made by humans, I acknowledge that my interpretations of events, artefacts, etc. are out of necessity my own and that other interpretations may exist. However, I have grounded my interpretations in such a way that they can always be traced back to the empirical data.

• Authenticity: Authenticity is important to convince the reader that the researcher was really doing fieldwork and understood “what is going on here” (Golden-Biddle & Locke, 1993). The text should demonstrate that the researcher knows the jargon and knows the population. Since I have been working in the study field for all my professional life I claim that this is indeed valid for me.

• Plausibility: This is about getting the reader to accept that the research contributes to the knowledge stock in the research area (ibid.). It’s a matter of getting the reader on your side: “I will go along with this... for the moment I suspend my disbelief”. This I have tried to achieve by reporting my results as well as possible.

• Criticality: This concerns the ability of the text to provoke the reader to reflect on his taken-for-granted assumptions of the research field: “By God, I never thought of that! “(ibid.). Again I have tried to achieve this by reporting my results as well as possible.

Another issue in action research is that the research aspect may be submerged. According to Checkland (1991) the most important issue that separates action research from mere action is the presence of an ‘intellectual framework’. Basically, all research includes a set of related ideas F which are used in a methodology M to investigate an area of research A. Action research will typically change or extend all these elements. The ‘intellectual framework’ F
CHAPTER 2

defines the research themes within which lessons can be learned. Unless the framework is declared, these lessons cannot be transferable to other areas of investigation.

In this study, the construct of activity domains renders the intellectual framework F. This means that the real world problems at Ericsson are analysed by using this construct as a frame of reference. The findings from this analysis are in turn modifying the intellectual framework F as actions are carried out in the application area.

2.1.5 Validity, reliability and relevance

Validity and reliability have been discussed frequently in interpretative research and there exist different interpretations of these concepts (Lundahl & Skärvad, 1982; Yin, 1989; Merriam, 1994; Svensson, 1996).

Validity

In the study I take the following position regarding validity:

Validity concerns the relation between the results of the study and its research questions. Is the research design set up and carried out in such a way that the results achieved are answers to the research questions? In other words, are we solving the right problem in the study?

As pointed out earlier, the Framework was introduced and used in the Ericsson practice before the research questions were formulated (the years before 1998). Thus, the research task must somehow be formulated in hindsight. Certainly, I had vague ideas about the feasibility of the Framework, but these were not articulated. This meant that one of the first tasks of my research was to articulate a preliminary version of my research questions. However, due to my postgraduate education and the scientific perspective in general, several other interesting aspects of the Framework emerged, for example its possible impact on information system development methods or organizational learning theories. Thus, my research questions were reformulated in a dialectical way as the properties of the Framework were unfolded.

Taken to its extreme this is certainly not in line with the research design that Yin (1989) suggests. In his more positivistic approach, the research design has to be redone if the research questions change. However, since the basic praxis perspective, which has informed the Framework from its outset, has not changed I claim that my research has basically refined this perspective. A vaguely formulated research design has been gradually transformed into an articulated one which is the one reported in the dissertation. The important aspect here is that the research questions, the empirical findings and the results are compliant with each other in the dissertation, which I claim they are.

Reliability

In the study I take the following position regarding validity:

Reliability concerns the quality of the research results. Something is reliable if it is worthy of reliance or trust. Are the claims well grounded? Is it possible to follow the argumentation? In other words, are we solving the problem right?
RESEARCH DESIGN AND REALIZATION

The reliability of the thesis is grounded in the following way:

- **Triangulation**: I have used several data sources: interviews, internal Ericsson documentation, personal notebooks, e-mail.
- **Contributions – peer reviewed**: Contributions to peer reviewed conferences and publications.
- **Contributions – not peer reviewed**: Presentation of my results at seminars, workshops, industry fairs, journals, etc. which are not peer reviewed.
- **Industrial grounding internally at Ericsson**: Presentation of my results internally at Ericsson.
- **Industrial grounding externally outside Ericsson**: Presentations and demonstrations at other companies than Ericsson. Contributions to my study from funding authorities.
- **Observation over long time**: The observation period stretches from the early 1990s until 2002, that is more than 10 years.
- **Make bias explicit**: This I have tried to do in section 1.6 My personal background.
- **“Follow the track”**: This means that the research should be carried out in such a way that another person is able to follow in detail how the information was collected, how the categories were developing and what decisions were made during the investigation. The ideal is some kind of manual from which the study can be repeated or replicated.

**Relevance**

The relevance of the study is confirmed by the usage of the Framework in the Ericsson practice. In 1998 the usage was zero. Late 1998 the first project started to use the Framework. From 1998 until mid 2002 about 140 main and subprojects have been impacted by the Framework, mostly as data providers of ongoing project coordination data. The most notable result is that the Framework is a key element in the coordination of the development of 3rd generation of mobile systems, which is the most important project at Ericsson at the moment (2002). More details about this can be found in Chapter 6 The history of the Framework.

In addition to this, the interest from other industrial settings and the scientific contributions indicate that the research questions and the knowledge contributions are relevant.

2.1.6 Generalizations from interpretative research

Generalizations from interpretative research differ from generalizations from quantitative studies. Yin (1989) coins the concept of **analytical generalization** for case study situations. This means that the theory gears the selection of generalizable domains. Findings should be generalized to theory using general categories that apply as well in other settings. This must be tested by replications (a second or third time) where the theory has specified that the same results should occur. Thus, generalization of results, from either single or multiple designs, is made to theory and not to populations as in quantitative studies.

Merriam (1994) means that generalization from one case can be done by redefining the concept of generalizability in the following ways:
• As a working hypothesis: Distinguish between situated factors and general factors when transferring from one situation to another. Supply a perspective rather than “truth”.

• As naturalistic generalizability: Humans look for patterns which can explain their experiences. Identify similarities in objects (and relationships, my comment) inside and outside the situation which guide actions.

• As concrete universals: The general can be found in the specific (manifestation in concrete ways of general structures).

• As a user / reader relationship: Let the user decide if the results in one situation are applicable in other situations as well.

Walsham suggests four types of generalizations of interpretative studies (Walsham, 1995, p. 79):

• The development of new concepts in order to capture new aspects of findings.

• The generation of theories in which new concepts may be part of a broad network or integrating clustering of concepts, propositions and world-views.

• Drawing of specific implications from in-depth case studies which can be seen as ‘generative mechanisms’ in other contexts. ‘Generative mechanisms’ are similar in nature to generative mechanisms in natural sciences. However, since social structures are dependent on human activity, generative mechanisms should be regarded as ‘tendencies’ which may provide explanations of past events but are not wholly predictive for future situations.

• The contribution of rich insight, that is, a rich empirical material may be read in several ways, thus giving rise to new insights.

The generalizations of the findings from this study are discussed in Section 9.4 Generalizing the results.

2.1.7 The artefact in science

Since the study includes the construction of an artefact, the Framework, a relevant issue is the role of science in this venture. This is discussed by Hendersen (1998). Science is trying to formulate the current state of affairs within a particular study field. Thus, it is primarily an analysis task. Design is trying to formulate a desired state of affairs, that is where we would like to be. Thus, it is primarily a task of imagining. Engineering is trying to answer how we should go from where we are to where we would like to be. Thus, it is primarily an
implementation task. Knowledge development may take place in all of these areas (see Figure 9).

In the study all these aspects can be found. The Activity Domain Theory and the praxis perspective, which has informed the construction of the Framework, may be seen as belonging to the scientific perspective. The design perspective may be characterized as a desired state where the consequences of the Activity Domain Theory have been implemented in a working support for the coordination of the development of complex systems. The engineering perspective is then the implementation of the Framework in the Ericsson development practice, using the Activity Domain Theory as guiding principles in the implementation.

However, it is important to recognize that the borders between these perspectives are not clear-cut and that the transition between the perspective is iterative in nature. This concerns knowledge developed in the dissertation as well as knowledge developed in the actual implementation of the Framework. In the first case the desired state (the design perspective) is the finished dissertation and in the latter a working implementation of the Framework. In both cases an alteration is taking place between the reflection phases end the action phases which might modify the desired state of affairs during the implementation.

2.2 Realization

In this section I describe how the research design was executed in the study.

2.2.1 Coordination domains at Ericsson

The Framework was used at four different coordination domains within Ericsson. These domains were established in the following order: the S-domain in Stockholm, Sweden, the A-domain in Aachen, Germany, the L-domain in Linköping, Sweden and the central C-domain in Stockholm, Sweden.

The S-domain:
The development focus of this domain was products included in the platform for the AXE system, that is, processors, switches, exchange terminals, systems for supervision and maintenance of the network, etc. The development scope included both hardware and software...
design. The number of projects impacted by the Framework was approximately 16 distributed to 7 development sites, including Australia, Italy, Croatia and Norway. The main foci for the coordination were change request management, requirement management and status accounting.

The A-domain:
This domain had three development foci. One was the Mobile Switching Centre node in the 3rd generation of mobile networks. Another concerned core switching functions in the mobile platform. A third was a node in the stationary telecommunication network. The development scope was mainly software design. The number of projects impacted was approximately 70 distributed to 24 different sites. The main foci for the coordination were requirement management, test management and work-package planning and integration.

The L-domain:
This domain developed various products in the base station area. The development scope was mainly software design. The number of projects impacted was around 20 distributed to 20 different sites. The main foci for the coordination were requirement management, change request management and status accounting.

The C-domain:
The main focus for this domain was the overall mobile core network, that is, rather than focusing on individual nodes as in the other domains, the C-domain focused on the network as a whole. Moreover, due to the centralistic ambitions during 2001 and 2002 this domain included various other projects. The number of projects was approximately 30. I have not been able to assess the number of sites impacted. The main focus for the coordination was the management of requirements from the network level to the node level.

In summary, approximately 140 projects distributed to more than 20 sites worldwide have been impacted by the four coordination domains.

2.2.2 Interviews
Altogether 18 interviews have been performed. The interviews were semi-structured in the sense that checklists were presented to the interviewees in advance. During the interviews the checklists were used as a reminder of items to be discussed. The reason for this choice is that I wanted to be as open as possible for discovering effects not anticipated. Each interview took 1-1.5 hours and was recorded on mini-disc.

All interviewees are anonymous and referred to by depersonalized identities in the study. All interviewees have also agreed to use quotations freely from the interviews.
The role and a short characterization of each interviewee are summarized in Table 5 below:

Table 5. Interviewee role and characterization

<table>
<thead>
<tr>
<th>ID</th>
<th>Role</th>
<th>Short characterization</th>
</tr>
</thead>
<tbody>
<tr>
<td>PM1</td>
<td>Project manager</td>
<td>About 25 years at Ericsson in testing, design, quality and project management. Also worked as product manager in various line management positions. Overall responsibility for business processes in the AXE-S project.</td>
</tr>
<tr>
<td>PM2</td>
<td>Project manager</td>
<td>Experienced project manager. Responsible for the first project that used the Framework.</td>
</tr>
<tr>
<td>PM3</td>
<td>Project manager</td>
<td>With Ericsson since 1993. Background in test management and configuration management. Quality coordinator. Project manager for test and configuration management projects in UMTS projects.</td>
</tr>
<tr>
<td>PM4</td>
<td>Project manager</td>
<td>Project manager in UMTS projects.</td>
</tr>
<tr>
<td>PM5</td>
<td>Project manager</td>
<td>Project manager and line manager. Worked also in the quality and process areas.</td>
</tr>
<tr>
<td>CU1</td>
<td>Customer representative</td>
<td>Background in computer software engineering, research and development. Evaluator of the AXE-S project concept as a customer representative.</td>
</tr>
<tr>
<td>MT1</td>
<td>Method &amp; Tools coordinator</td>
<td>14 years with Ericsson. Extensive experiences in methods and tools for telecommunication systems development.</td>
</tr>
<tr>
<td>MT2</td>
<td>Method &amp; Tools coordinator</td>
<td>Tool responsibility and eMatrix champion at the local development site in Australia for the first project that used the Framework.</td>
</tr>
<tr>
<td>MT4</td>
<td>Method &amp; Tools coordinator</td>
<td>eMatrix champion for development projects at the Danish development site using the A-domain.</td>
</tr>
<tr>
<td>MT5</td>
<td>Method &amp; Tools coordinator</td>
<td>Tool responsibility at the A-domain.</td>
</tr>
<tr>
<td>IT1</td>
<td>IT expert</td>
<td>With Ericsson since 74, worked with methods and tools with a focus on information systems.</td>
</tr>
<tr>
<td>IT2</td>
<td>IT Manager</td>
<td>33 years with Ericsson. Corporate responsible for product data management and information systems in the late 90s.</td>
</tr>
<tr>
<td>DE1</td>
<td>Designer</td>
<td>Worked at Ericsson between 1993-2000 as a designer and developer of methods and tools.</td>
</tr>
<tr>
<td>CM1</td>
<td>Configuration manager</td>
<td>Configuration manager since 1995. CM responsible in the first project that used the Framework.</td>
</tr>
<tr>
<td>CM2</td>
<td>Configuration manager</td>
<td>With Ericsson since 1997. Worked together with CM1 in the first projects that used the Framework.</td>
</tr>
<tr>
<td>VDR1</td>
<td>IS vendor consultant</td>
<td>Worked as a consultant for the eMatrix tool vendor in the Nordic countries since 1997. Main responsibility for the technical tool implementation at Ericsson since 1998.</td>
</tr>
</tbody>
</table>

2. A contact person between the project and the technical implementors of the eMatrix tool. Has a very important role in getting the tool accepted and used in the early projects.

Whenever a quotation is used in the dissertation it is referred to as: role and recorded time in the interview for the paragraph where the quotation is made, for example (VDR1-006:36).
Interviews for research question RQ3

The interview checklist for the research question RQ3: “Which elements in the Ericsson development practice contribute to successful outcomes of development projects?” is summarized in Table 6. The people interviewed were PM1, CU1, MT1, IT1 and DE1.

Table 6. Interview questions for research question RQ3

<table>
<thead>
<tr>
<th>Development project?</th>
<th>Your role?</th>
<th>Background?</th>
<th>Success factors?</th>
<th>Problems?</th>
</tr>
</thead>
<tbody>
<tr>
<td>- lead time (months)</td>
<td>- customer</td>
<td>- experience</td>
<td>- parallel development</td>
<td>- overview</td>
</tr>
<tr>
<td>- size (man-hours)</td>
<td>- project mgr</td>
<td>- education</td>
<td>- architecture target system</td>
<td>- design</td>
</tr>
<tr>
<td>- persons (number)</td>
<td>- line mgr</td>
<td></td>
<td>- requirements</td>
<td>- requirements</td>
</tr>
<tr>
<td>- geographical distr</td>
<td>- designer</td>
<td></td>
<td>- project organization</td>
<td>- information mgmt</td>
</tr>
<tr>
<td>- type (new, traditional)</td>
<td>- tester</td>
<td></td>
<td>- project mgr</td>
<td>- information systems</td>
</tr>
<tr>
<td>- result (success, failure)</td>
<td>- methods &amp; tools</td>
<td></td>
<td>- environment</td>
<td>- processes and methods</td>
</tr>
<tr>
<td>- technology (new, well-tried)</td>
<td>- system mgr</td>
<td></td>
<td>- transparency</td>
<td>- teamwork</td>
</tr>
<tr>
<td></td>
<td>- configuration manager</td>
<td></td>
<td>- communication</td>
<td>- organization</td>
</tr>
<tr>
<td></td>
<td>- requirement manager</td>
<td></td>
<td>- integration planning</td>
<td>- project model</td>
</tr>
</tbody>
</table>

In addition to this, data from the other interviews was used whenever appropriate for this research question. Thus, the data for RQ3 is based on focused investigations using the questions in Table 6 and data from questions for research question RQ2 (Table 7).

Interviews for research question RQ2

The interview checklist for research question RQ2: “What are the overall consequences from the Framework intervention in the Ericsson development practice?” is summarized in Table 7. The basic question to the interviewees was:
RESEARCH DESIGN AND REALIZATION

“You are / have been participating in a development task / project where the management framework (or parts of it) has been used. What effects have you noticed that might be related to the framework?”

Table 7. Interview questions for research question RQ2

<table>
<thead>
<tr>
<th>Measurable effects?</th>
<th>Non-measurable effects?</th>
<th>Effects on understanding?</th>
<th>Other effects?</th>
</tr>
</thead>
</table>
| - effort (time) for doing tasks  
- cost for doing tasks  
- global access to information simultaneously  
- project planning, monitoring and co-ordination  
- management of incremental development  
- change request management without documents  
- requirement management without documents  
- information search including full text search  
- automatic generation of reports | - tasks previously not possible to do  
- the flexibility of the information system, doing many changes  
- having one information system for managing items rather then several, one common object model  
- complexity of doing tasks  
- traceability, both between items and in time  
- transparency, understanding how items are related, causes and effects  
- dependencies between items  
- having the management information in one tool | - understanding of the development task  
- of getting new insights  
- new ways of working  
- contribution to consensus or disagreements | - interaction with other eMatrix sites  
- resources, efforts  
- conflicts  
- performance  
- discussions about elements common to all projects or local to your project |

The questions have been formulated with the structure and properties of coordination domains in mind. For example, questions concerning understanding are focusing on the intersubjectivity constituent of coordination domains.

Table 8 indicates which people have been interviewed, how they are distributed to the different sites across Ericsson and which domain (usage or coordination) was in focus.

Table 8. Distribution of interviewees for research question RQ2

| Interviewee / Domain focus | Domain focus | S-domain, Stockholm | A-domain, Aachen | L-domain Linköping, | C-domain, Stockholm |
|---------------------------|--------------|---------------------|------------------|---------------------|
| PM2                       | usage        | Sweden              | Italy            | Australia           |                    |
| CM1                       | usage        |                     |                  |                     |                    |
| CM2                       | usage        |                     |                  |                     |                    |
| MT2                       | usage        |                     |                  |                     |                    |
| PM3                       | usage        |                     |                  |                     |                    |
| PM4                       | usage        |                     |                  |                     |                    |
| PM5                       | usage        |                     |                  |                     |                    |
| MT3                       | coordination |                     |                  |                     |                    |
| MT4                       | coordination |                     |                  |                     |                    |
| MT5                       | coordination |                     |                  |                     |                    |
| VDR1                      | coordination |                     |                  |                     |                    |
CHAPTER 2

The selection of interviewees is a compromise between available resources, time frame for the dissertation and an ambition to cover as many aspects of the Framework as possible. The focus of the interviews is on the S-domain in Stockholm and the A-domain in Aachen. At the S-domain the application was focused on requirement management, change management and project control. At the A-domain the focus was on coordination of the integration of software development packages (work packages). Thus these domains spanned different parts of the coordination context. Both the L-domain and the C-domain applications were essentially the same as the one at the S-domain.

The distribution of interviewees tries to capture the dimensions of geographical distribution, type of coordination and type of development project. I also have tried to capture the most important roles in the selection of interviewees, both from coordination domain focus and the usage domain focus. The only role which covers all applications is VDR1 which may be somewhat surprising. However, this is a consequence of the relative autonomy of the different activity domains and the lack of central coordination of the individual domains. There is at present (2002) no corporate responsibility for coordinating the various applications.

Additional interviews
In addition to the interviews above, IT2 was interviewed twice concerning the information system aspects of the Framework. As responsible for Corporate IT strategies he was a strong promoter of the PDM system chosen as the backbone for product management at Ericsson. However, he was also interested in the Framework approach although he did not endorse the choice of eMatrix as the IS in the Framework. eMatrix was a competitor to the Corporate PDM system on the commercial market. From the Corporate point of view, another PDM system within Ericsson was against Corporate strategies and a potential threat to the deployment of the global backbone.

Feedback from informants
The transcripts of the interviews have been provided to the informants in order to comment on these. In the transcripts I have enclosed my comments which means that the informants have been able to react to my interpretations. The knowledge contribution from research question RQ1 “How did the Framework evolve in the Ericsson development practice?” has been discussed with the informant PM1 who has confirmed the validity of the findings. The other knowledge contributions have been validated through feedback on my interpretations in the transcripts. However, I have not been able to get feedback from informants PM3, MT5 and DE1 due to the turbulent situation at Ericsson. For DE1 I have used, in addition to the interview, a written summary of his experience at Ericsson.

2.2.3 Data triangulation
Several sources of data have been used: nearly 500 interview statements, about 270 Ericsson internal documents, 44 personal notebooks between 1991-2002, about 10,000 e-mails between 1993-2002, personal calendars between 1990-2002. The notebooks include notes from meetings, my personal reflections, etc. These notes were, however, not collected as part of the research design since the research started in 1998.
2.2.4 Grounding in industrial settings

The grounding in industrial settings outside the Ericsson company has been done in several ways. The work has been sponsored by the Swedish National Board for Industrial and Technical Development, NUTEK, project P10518-1. The results were evaluated at a hearing in 2000. Furthermore, parts of the study have been presented at the following industrial seminars (most by invitation, not peer-reviewed):

Table 9. Contributions – not peer reviewed

<table>
<thead>
<tr>
<th>Seminars, presentations, etc. (not peer reviewed)</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technia-dagen</td>
<td>Seminar PDM, Saltsjöbaden, 1998</td>
</tr>
<tr>
<td>NUTEK seminar</td>
<td>1998</td>
</tr>
<tr>
<td>Future of PDM Global PDM – Web Enabled in Software- and Hardware Development</td>
<td>PDM-summit, Finland, IIR (invited key-note speaker), 1999</td>
</tr>
<tr>
<td>e99 SINTEF Norge</td>
<td>Oslo, 1999</td>
</tr>
<tr>
<td>A Global Web-based PDM support for Developing Complex Telecommunication Systems</td>
<td>CIM-data, Nice, 1999</td>
</tr>
<tr>
<td>NUTEK seminar</td>
<td>Poster, Luleå, 1999</td>
</tr>
<tr>
<td>How Ericsson works with Integration Driven Development</td>
<td>IIR CM seminar, Stockholm, 1999</td>
</tr>
<tr>
<td>Erfarenheter från användning av PDM-systemet eMatrix i Ericssons produktutveckling</td>
<td>Norrköping, seminar, 1999</td>
</tr>
<tr>
<td>Flexibilitet och kreativitet i komplexa projekt</td>
<td>Norrköping Nysum, 2000</td>
</tr>
<tr>
<td>NUTEK evaluation meeting</td>
<td>Stockholm, 2000</td>
</tr>
<tr>
<td>Soft 23:To Represent and Make Visible Concepts and Relations</td>
<td>Norrköping, 2001, seminar with John Edwards</td>
</tr>
<tr>
<td>IEEE Mgmt: Some thoughts about Organisational Knowledge</td>
<td>KTH, Stockholm, 2001</td>
</tr>
</tbody>
</table>
CHAPTER 2

The study has also been referred in the following publications:

Table 10. Publications – not peer reviewed

<table>
<thead>
<tr>
<th>Title</th>
<th>Publication</th>
</tr>
</thead>
<tbody>
<tr>
<td>“ASQ is more than testing tools”</td>
<td>Application development trends, USA, interview. Jan 2000 issue,</td>
</tr>
<tr>
<td>“Ökad kvalitet” Ericsson satsar på eMatrix</td>
<td>Verkstadsforum, bilaga 2/2000, pp. 18-19</td>
</tr>
<tr>
<td>“Organization Learning Revisited”</td>
<td>HW Designer (internal Ericsson journal), 2002</td>
</tr>
</tbody>
</table>

2.2.5 Industrial grounding internally at Ericsson

A substantial part of the deployment of the Framework included selling in the new concept to the Ericsson organization and thus grounding it internally. This was especially hard since the Framework was not an initiative coming from top-management. The sell-in was mainly done by demonstrations of the Framework on a laptop where the IS was implemented. In the table below the most important events are summarized. It can be seen that a large number of demonstrations took place during 1998. This was a critical period since the Framework initiative had been halted. During one of these demonstrations, PM2 became interested in using the Framework, something which eventually lead to the current (2002) widespread use of it (see Chapter 6 The history of the Framework).

In Table 11 the internal grounding at Ericsson is given. The three-letter acronyms are internal denotations of line units.

Table 11. Internal grounding at Ericsson

<table>
<thead>
<tr>
<th>Ericsson events</th>
<th>Date</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ericsson Rational software vendor demo</td>
<td>10 Feb 1998</td>
<td>Before the Rational software vendor take over of the SW-development responsibility at Ericsson</td>
</tr>
<tr>
<td>Ericsson UAB/O steering group demo</td>
<td>04 Mar 1998</td>
<td>Own dept. steering group</td>
</tr>
<tr>
<td>Ericsson UAB/PMO demo</td>
<td>05 Mar 1998</td>
<td>Project mgmnt network</td>
</tr>
<tr>
<td>Ericsson UAB/B/O demo</td>
<td>06 Mar 1998</td>
<td>Project mgmnt processor development dept.</td>
</tr>
<tr>
<td>Ericsson UAB/l demo</td>
<td>20 Mar 1998</td>
<td>Dept. of I/O units AXE systems</td>
</tr>
<tr>
<td>Ericsson PAC demo</td>
<td>27 Mar 1998</td>
<td>Managers methods &amp; tools development AXE systems</td>
</tr>
<tr>
<td>Ericsson LSO conference demo</td>
<td>02 Apr 1998</td>
<td>Local Support Organisation Ericsson</td>
</tr>
<tr>
<td>Ericsson NY/ETX demo</td>
<td>17 Apr 1998</td>
<td>Development dept., Nyköping</td>
</tr>
<tr>
<td>Ericsson UAB CM demo</td>
<td>23 Apr 1998</td>
<td>Conf. managers UAB</td>
</tr>
<tr>
<td>Ericsson UAB/X demo</td>
<td>07 May 1998</td>
<td>Switching equipment development AXE system</td>
</tr>
<tr>
<td>Ericsson CMcc conference</td>
<td>05 May 99</td>
<td>10 min. presentation</td>
</tr>
<tr>
<td>The Future of PDM – A Global Web-based Support for Hardware- and Software Development</td>
<td>21 Sep 1999</td>
<td>Hardware Design Seminar, HWDS, Helsinki, Finland</td>
</tr>
</tbody>
</table>
2.2.6 Industrial grounding externally outside Ericsson

The Framework has been presented to the following Nordic companies, mainly on the initiative of the eMatrix vendor in the Nordic countries, Technia (see Table 12). This has been done either as reference visits in place at Ericsson or at the location of the company:

Table 12. Industrial reference visits

<table>
<thead>
<tr>
<th>Industrial visits</th>
<th>Business</th>
<th>Country</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>NERA demo</td>
<td>Satellite equipment development &amp; production</td>
<td>Norway</td>
<td>1998</td>
</tr>
<tr>
<td>IVF demo</td>
<td>Industry cooperation institute</td>
<td>Sweden</td>
<td>1999</td>
</tr>
<tr>
<td>Sonera demo</td>
<td>Telecommunication operator</td>
<td>Finland</td>
<td>2000</td>
</tr>
<tr>
<td>Allgon demo</td>
<td>Antenna equipments for mobile communication</td>
<td>Sweden</td>
<td>2000</td>
</tr>
<tr>
<td>Whirlpool demo</td>
<td>Microwave oven development &amp; production</td>
<td>Sweden</td>
<td>2000</td>
</tr>
<tr>
<td>Valmet demo</td>
<td>Paper mill and forest machinery manufacturer</td>
<td>Finland</td>
<td>2000</td>
</tr>
<tr>
<td>Mölndlycke demo</td>
<td>Medical equipment manufacturer</td>
<td>Sweden</td>
<td>2000</td>
</tr>
<tr>
<td>Scania, presentation</td>
<td>Global truck manufacturer</td>
<td>Sweden</td>
<td>2000</td>
</tr>
</tbody>
</table>

2.2.7 Research processes, strategies and methods

In this section I will describe the research processes, strategies and methods used in the study.

Research processes and strategies

In Section 2.1.2 the different roles of theories in research processes were discussed. The role of theories and strategies to obtain the expected knowledge contributions in this study can be described as follows.
The first research question RQ1 “How did the Framework evolve in the Ericsson development practice?” provides reconstructive / historical knowledge. No particular role of theory can be found in connection with this research question. The strategy to provide the knowledge is a historical / reconstructive one\(^3\).

The second research question RQ2 “What are the overall consequences from the Framework intervention in the Ericsson development practice?” provides explanatory knowledge concerning the consequences of the Framework intervention. In this sense it can be regarded as a theory generating approach, albeit not aiming at a full-fledged theory. The perspective of Activity Domain Theory is used only as a sensitizing device in analysing the data. The research process for this research question can be characterized as mainly an inductive one. The strategy to provide this knowledge is theory generation grounded in the Ericsson empiri.

The third research question RQ3 “Which elements in the Ericsson development practice contribute to successful outcomes of development projects?” provides explanatory knowledge. Here, the initial theory permeation is low and can best be characterized as a framing of the research interest by identifying the unit of analysis as the ‘project’. The result of RQ3 will be a higher theory permeation providing insights about the elements contributing to successful outcomes. The theory generation takes place in several oscillations between theory and data collection (see Chapter 8 Reinforcement rods). Thus it can be characterized as an abduction process. The strategy to provide this knowledge is theory generation grounded in the Ericsson empiri.

Concerning the fourth research question RQ4 “What are the impacts on coordination from the Framework intervention in the Ericsson development practice?”, there is a high theory permeation at the outset consisting of the Activity Domain Theory which in turn is derived from the praxis perspective (see Chapter 3 The Activity Domain Theory). This perspective has been informing both the construction of the Framework as well as the data collection and the analysis of the data. Thus, this research question can be regarded as a validation of the constructive power of this theoretical perspective w.r.t. the coordination of the development of complex systems. This means that the research process related to this research question can be characterized as a theory validating process, i.e. a deductive one.

---

\(^3\) The classification of strategies comes from Goldkuhl (1998, my translation).
In Figure 10 the different research processes involved in the study are illustrated:

![Figure 10. Research processes in the study](image)

**Method**

The method used for the research questions RQ2, RQ3 and RQ4 is basically a modified Grounded Theory (GT) approach. The reason why I did not use GT as is (Strauss & Corbin, 1998), is that I find it vague on the point of using theory as a “searchlight” for collecting the empiri. The ideal seems to be that the researcher enters the research field without any preconceptions about it, trying to avoid being biased. The “...actual properties emerge from data...” (ibid., p. 96). I do not consider this to be a realistic option.

Regarding research question RQ3 a preliminary analysis was done in connection with a graduate course, in which five reinforcement rods were identified: participation, focus, orientation, federation and core. Here, the GT method was followed more strictly. However, the development of properties and dimensions of each category was simplified. This means that a theoretical saturation of each category was not achieved, which in turn implies that a more thorough analysis may reveal other categories or a better articulation of the existing ones. However, since RQ3 is not the main focus of the study, I claim that this simplification of the GT approach is justified.

The result from this analysis was used to articulate the definition of coordination. Moreover, the categories found were further grounded in data from projects where the Framework was used. Thus, I claim that the categories found are well grounded although not exhaustive.

For research question RQ2 and RQ4, the Activity Domain Theory was used as a theoretical guidance. Furthermore, the Framework itself was utilized in the analysis work which means...
that I have defined an activity domain for the analysis. The conceptual model used is shown in Figure 11.

The analysis follows a dialectical knowledge production process. In this process the analysis moves from the concrete and situated over to the abstract and general and then back to the concrete and situational. This indicates that there are two types of analysis categories: empirical and theoretical. The empirical categories are those related to the concrete empir. These categories try to capture a rich and situated totality in which I also include the Framework elements for RQ2 and RQ4 (Chapter 5 The structure of the Framework). An empirical category, for example ‘product’ is apprehended as a context rather than an isolated category. In order to characterize this context as fully as possible characteristic actions related to that category are considered. These actions are described in the form of GT: prerequisites (called ‘conditions’ in GT), actions and consequences (Strauss & Corbin, 1998, p. 128 ff.). The combined static and dynamic characterization is in line with the GT coding process: “Coding for process occurs simultaneously with coding for properties and dimensions and relationships among concepts” (ibid., p. 167).

The next step in the analysis is to move from the concrete and situational to the abstract and general. In this step theoretical categories, which are not specific for the particular situation, are identified. These categories are of several types. One type concerns the ‘reinforcement’, e.g. categories which correspond to RQ3. The other type concerns the coordination constituents in the Activity Domain Theory, i.e. intersubjectivity, contextuality, domain transition, experiential learning, orientation, temporality, stabilizing core, tool, motive, object and outcome. These categories are also grounded in references in the literature. Finally I have included the axial and selective coding categories of GT.

Figure 11. Conceptual model used in the analysis
The last step in the analysis is to return from the abstract to the concrete, hopefully with a deeper understanding of the concrete situation. This is done by seeing empirical findings as a ‘manifestation’ of the theoretical, abstract categories. Thus the analysis is circular in going from the concrete to the abstract and back to the concrete again. This movement may be loosely characterized as theory generating inductive approach followed by a deductive, hypothesis grounding approach.

The analysis was carried out in the following way: The conceptual model in Figure 11 was implemented in eMatrix. For each empirical source the following steps were performed:

1. Create an instance of the empirical source (quotation) and give it a unique identity. Relate it to the source by the relation ‘included_in’.
2. Identify the most salient effect category of the quotation. This will identify a context in which to place the quotation.
3. Find empirical categories (‘effect’ or ‘Framework’) and relate it to the data by the relation ‘grounded_in’. Whenever the effect and Framework categories are similar, use the Framework category if appropriate. Note actions in the form of Prerequisites (P), Actions (A) and Consequences (C) if appropriate. Add an interpretation of the quotation. The steps 1 to 3 correspond roughly to the open coding step in GT.
4. Generate theoretical categories and relate them to the empirical ones. This corresponds roughly to the axial and selective coding steps in GT.

As an example, interviewee PM4 stated:

“I am now working at a Wireline project also on the AXE, if you are to see the differences between the Wireline and the mobile having exactly the same technology so to say. That is already an interesting challenge to try to align that, if you don’t even try to go beyond platforms and so on, forget it. Not the first common fight I would take [...] you would have to come up with a system that is so generic that nobody can really use it [...] nobody is really happy with the results.” (PM4-11-1:20c)

In Figure 12 the effect categories, which the quotation grounds, are shown together with my interpretation of the grounding.

<table>
<thead>
<tr>
<th>Effect</th>
<th>Description</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>quotation(PM4-11-1:20c)</td>
<td>I am now working at a Wireline project also on the AXE, if you are to see the differences between the Wireline and the mobile having exactly the same technology so to say. That is already an interesting challenge to try to align that, if you don’t even try to go beyond platforms and so on, forget it. Not the first common fight I would take [...] you would have to come up with a system that is so generic that nobody can really use it [...] nobody is really happy with the results.</td>
<td></td>
</tr>
<tr>
<td>Interview PM4</td>
<td>Project manager in UMTS projects.</td>
<td></td>
</tr>
<tr>
<td>aCPR</td>
<td>agent which comes from the CEP activity domain.</td>
<td></td>
</tr>
<tr>
<td>effect category / shared meaning</td>
<td>Tissues related to understanding interpretation and meaning.</td>
<td>There is a need to align the model (P). However, the effort to do this is very large (E). Thus the should not be done (C).</td>
</tr>
<tr>
<td>effect category 15 federations</td>
<td>Tissues regarding one or several data bases in Matrix.</td>
<td>If these parties cannot agree, the lack of several databases increases.</td>
</tr>
<tr>
<td>effect category 1</td>
<td>especially interesting quotations.</td>
<td>Captures the shared understanding problem mainly.</td>
</tr>
<tr>
<td>effect category information model</td>
<td>The conceptual model refers to an information model, business model, object model, product model etc.</td>
<td>Even if there is some technology which has a strong architectural component the back and the designed. This is the core observation that H1A made. One reason why some common data base will not work is that line going to daily usage is necessary. The effort to coordinate this is too high.</td>
</tr>
</tbody>
</table>

Figure 12. An example of data analysis
CHAPTER 2

In Table 13 the research design is summarized:

Table 13. Summary of the research design

<table>
<thead>
<tr>
<th>Detailed research question</th>
<th>Expected knowledge contribution</th>
<th>Knowledge type</th>
<th>Strategy</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>RQ3: Which elements in the Ericsson development practice contribute to successful outcomes of development projects?</td>
<td>Identified elements</td>
<td>Explanatory</td>
<td>Explorative, theory generation on empirical grounds</td>
<td>Modified Grounded Theory, low theory preconception</td>
</tr>
<tr>
<td>RQ4: What are the impacts on coordination from the Framework intervention in the Ericsson development practice?</td>
<td>Identified impacts on coordination</td>
<td>Explanatory</td>
<td>Explorative, theory generation on empirical grounds</td>
<td>Modified Grounded Theory, high theory preconception (the Activity Domain Theory)</td>
</tr>
</tbody>
</table>

2.2.8 Restrictions

All interviewees agreed to let me quote from the interviews without restriction. The only restriction from Ericsson is that sensitive information regarding financial and project lead time data must be withheld. The text in the dissertation has been checked with my mentor at Ericsson. My judgement is that these restrictions have not impacted the findings in a significant way.
PART II - THEORY AND DOMAIN CONSTRUCTION

In Chapter 3 The Activity Domain Theory the theoretical grounding of the framework is treated. The chapter starts with an account for the praxis perspective. Based on this perspective, a set of basic assumptions concerning human activity is defined. Next we discuss a semiotic perspective of the interaction between human actors and their environment. From this, a stratified model of the interaction is described which consists of a connectional, conceptual and linguistic strata. From the basic assumptions a set of coordination constituents is conjured which we claim are fundamental for coordinating human activity. These constituents are grounded in the cognitive, semiotic and linguistic literature by analyzing the stratified interaction model for each constituent. From the coordination constituents the structure of activity domains is defined.

In Chapter 4 Articulating Coordination we discuss an elaborated conception of coordination as an activity domain with the purpose of providing coordination to activity domains such as product development projects.

In Chapter 5 The Structure of the Framework the Framework and its elements are described. These elements are derived from the coordination constituents which means that the Framework is apprehended as the appropriation of the Activity Domain Theory for the purpose of coordinating complex systems’ development.
3 The Activity Domain Theory

“There is nothing so practical as a good theory.” Kurt Lewin

In this chapter we will propose a theory for socio-human activity which we call the Activity Domain Theory\(^1\). This chapter is a substantial and less accessible part of the dissertation and one might wonder why we have put this much emphasis on the theoretical perspective. There are several reasons for this. One reason is that we claim that the theory is a contribution in itself, which may be used as an alternative theoretical foundation for explanatory and constructive purposes related to coordination. Since the practical management of coordination presumes some kind of information system (IS) support, this theoretical foundation may also be used to erect IS on (see Iivari & Lyytinen, 1998). The results of the study show that this is indeed the case. Another reason is that there is a need to explain the ideas behind the Framework. Why is it structured the way it is? How come that there are three models and one information system in the Framework?

The theory originates in the praxis perspective and tries to encompass both individual / subjective aspects as well as social / objective aspects of human activity. In particular, the theory emphasizes the dialectical relation between the individual and society as the genesis of human capabilities and knowledge. The reality as perceived by actors is mediated by signs which have a physical signifier aspect and a signifying aspect. The signifier and the signified are associated in the brain of humans. Thus, our theory presumes that there exists an objective world which is independent of, but related to human beings. However, human activity changes that world which in turn changes the perceived reality of humans.

We will propose a set of coordination constituents which we argue is fundamental for coordinating socio-human activity. We proceed to ground these constituents in the cognitive and linguistic literature, thus underpinning their fundamental character. These constituents are utilized in the structuring of praxis into activity domains, which we consider to be a fundamental form in which the coordination of socio-human activity can be moulded. The Framework, which is derived from the activity domain construct, provides a bridge between the theory and the operational support for coordinating the development of complex systems.

From our understanding of coordination as described in Chapter 4 Articulating Coordination, it is clear that coordination can be regarded as a certain form of human activity. This means that we must consider human beings and their actions, meanings, interpretations, intentions, etc. in the context of coordination. Thus, the individual and subjective are important aspects of coordination. On the other hand, coordination is done by several actors working together. This implies that social issues like shared meaning or intersubjectivity are important. Furthermore, the coordination task is supported by certain tools, procedures, rules, conventions, etc. which are not dependent on any particular individual. Thus there is a trans-individual, objective aspect of coordination.

In order to study human activity we must proceed from some assumptions about the nature of knowledge, man, society and the world (for a discussion about these issues, see e.g. Israel, 1980). These basic assumptions must be made explicit, since they are guiding princi-

\(^1\) Our understanding of ‘theory’ is discussed in Section 2.1.2 The roles of theories (p. 26).
CHAPTER 3

amples for how the researcher apprehends the problem under investigation and what methods are used. In scientific discourse the basic assumptions, or, in Lakatos’ terms ‘hard core’ of the theory (see e.g. Chalmers, 1976), may be discussed in terms of pros and cons in relation to problem to be investigated. However, the hard core cannot be evaluated in terms of being true or not. The validity of the chosen hard core lies in its ability to provide interesting, relevant and well-grounded research results. Once this ceases, the hard core should give way to other basic assumptions, which then provide a new hard core for guiding further research.

The nature of human activity has been a basic theme in philosophical discourse for centuries. We will of course not dive into the depths and subtleties of this discourse here. In general, the constituents of human activity have been seen either as individual action or social structures. However, some philosophers have focused on the interaction between the individual and society, for example Marx (1973/1857), Berger & Luckmann (1966), Kosík (1976) and Giddens (1984) to mention but a few.

Some of the theories proposed have been appropriated to inform the construction of information systems (IS), for example, the Structuration Theory of Giddens (Jones, 1999). However, there are few practical and useful results reported (Iivari & Lyytinen, 1998). Other theoretical approaches are the Actor Network Theory (Latour, 1998), Activity Theory (Engeström, 1999) and Organisational Semiotics (Stamper, 2001). Again, the practical results which meet the demands of global IS support in our application area seem to be limited.

Line of reasoning
The line of reasoning is as follows (see Figure 13).

We approach the Activity Domain Theory from two directions. The first one is praxis as understood in the Marxian tradition (Kosík, 1976; Israel, 1980). Praxis has a total view of human activity where the individual and social are formed in continuous interaction with each other. This makes praxis suitable as a starting point for our purposes. A thorough analysis of alternative perspectives is beyond the scope of this thesis. However, a superficial

Figure 13. The line of reasoning
THE ACTIVITY DOMAIN THEORY

analysis reveals that most alternatives emphasize one or several aspects of praxis but in general not all of them (see Section 3.5 Alternative perspectives on human activity).

From an analysis of praxis Section 3.1 The praxis perspective we will establish a set of basic assumptions about human activity. From these assumptions we define a set of coordination constituents which make up the core of the Activity Domain Theory. We conjecture that these constituents are fundamental for coordinating human activity:

• intersubjectivity
• contextuality
• domain transition
• experiential learning
• orientation
• temporality
• stabilizing core
• tool usage
• motive, outcome, object

However, in order to ground this conjecture theoretically we will approach the Activity Domain Theory from a semiotic and cognitive direction as well (Section 3.2 Signs as mediators of the dialectical relation). In praxis the dialectical relation is regarded as the genesis of human knowledge. In order to concretize the dialectical relation we elaborate on a stratified model for the cognitive human system given by Gärdenfors (2000a). Rather than locating the model in the individual, as Gärdenfors does, we apprehend it as a model for the dialectical relation in which the interaction between the individual and its environment is mediated by signs.

The stratified model consists of three strata: the linguistic, conceptual and connectional ones. These strata capture social aspects of humans (language) as well as individual aspects (concept formation and connectivity of neurons in the brain). For each coordination constituent we will analyze how the constituent can be grounded in the three strata. If a constituent can be well-grounded in any or several of the strata, we argue that it is fundamental for human activity.

In the analysis we will mainly refer to the works of Kosík (1976) and Israel (1980) regarding the praxis philosophy, Gärdenfors (2000a) regarding cognitive issues and Vološinov (1986/ 1929) regarding semiotic and linguistic issues. We will also use some constructs from the Activity Theory (Engeström, 1999; Kuutti, 1991). The reason why we have chosen to ground the Activity Domain Theory in these works is that they, according to our opinion, provide far-reaching insights about major issues concerning this theory.
3.1 The praxis perspective

“God does not work, though he creates, but man both creates and works” Saying from the Renaissance

In the Marxian tradition, praxis is the central concept where man as a social being transforms his reality and in turn is transformed by that reality. The strong point about the praxis concept is its totality, that it tries to encompass the whole of human activity. This makes it an attractive theoretical point of departure for dealing with the coordination problem. The weak point is that praxis, as understood by Marx, is a nebulous and difficult concept which has been discussed, criticised and debated many times. The thoughts of the young Marx and Engels were also severely distorted during the Stalinist era in the Soviet Union. Moreover, there are few if any practical results reported in the literature which base their approach on praxis and it is hard to see how it can be structured to the point where it can be appropriated into normative knowledge for system design and analysis.

Here, we will not engage in philosophical inquiries about the praxis concept as such. Our purpose is more modest. We will utilize the praxis category as a fundamental perspective for chiselling out our approach to the coordination task:

“Human praxis [is] the arena for the metamorphosis of the objective into the subjective and of the subjective into the objective.” (Kosik, 1976, p. 71).

3.1.1 Ontology

The first important issue concerns ontology, i.e., what reality is or “what there is” (Israel, 1980, p. 2). What does reality consist of and how is it structured? What is the nature of the universe? What is the place of man in reality?

One possible ontology is the dualistic understanding in which material and psychological categories are understood as clearly separated and related. Their relation to each other is external, which means that the categories have nothing in common other than that they happen to be juxtaposed in a certain context, for example a car being parked at a certain parking place. When two things are related externally, they are not changed by the relation. Examples of such categories are ‘matter’ versus ‘spirit’, ‘subject’ versus ‘object’, etc. Language is apprehended as external to the world, as a system of signs that describes the world and helps us to manipulate it. This understanding dates back to Descartes who asserted that body and soul were two quite different substances. A modern variant of the dualistic understanding can be found in, for example, Habermas ‘world relations’ between the speech act and the ‘subjective’, ‘objective’ and ‘social’ worlds respectively (Habermas, 1984).

---

2. ‘Praxis’ comes from the greek verb prasso, which means roughly “action”. Praxis was used by Aristotle to denote actions which are goals as such. The ability to realize such goals is phronesis or wisdom. Poiesis denote actions which are realizing goals outside the actions. The ability to realize these goals is called techne. Finally, abilities which are a prerequisite for theorica, theoretical / scientifical knowledge, is called episteme (see for example Liedman, 1998, p. 266).
Another possible understanding of the world is monism which denies the existence of separate categories of things or phenomena. Monism finds one where dualism finds two. Monism claims that everything psychological can be reduced to physiological and eventually material categories, i.e., monism represents an atomistic way of thinking. For example, psychology can be expressed in the language of biology which in turn can be expressed in the language of physics, etc. Atomism tries to build a system by arranging, categorising and correlating basic building blocks. The opposite to atomism is holism. In holism, the totality is considered as something that ‘emerges’ spontaneously and which conveys new properties to the parts, that is, the whole has precedence over the parts.

An alternative ontology is the totality category in dialectical thinking. As in dualism there are pairs of opposites. However, the relationship between these pairs is internal, which means that opposites are different but mutually depending on and impacting each other within a totality. The elements in an internal relation are defined by the relation, and all elements are necessary to form the totality. The elements do not work or exist independent of each other. An element in an internal relation cannot be defined without referring to the others. An example of this type of relation is the ‘master’-‘slave’ relation. The meaning in the word ‘subject’ is defined by referring to the word ‘object’ and vice versa. The totality is more basic than the elements taken one by one.

The totality category negates dualism, the separation of the world in distinct, externally related categories. It also negates monistic reduction and atomism. Furthermore it negates holism, which is in contrast to the dialectical thinking where totality is regarded as both producing and being produced in an ongoing process (Israel, 1980, p. 96). There is no precedence of parts over the whole or the other way around.

Praxis can be seen as the process through which man produces the totality of his world, obliterates what is given and transcends it. This means that the world exists independent of the subject but not external to the subject. The knowledge we as human beings have about this reality is intrinsically related to the cognitive capabilities of man. However, this does not mean that our knowledge is subjective and private. On the contrary, every individual understanding of the world presupposes the presence of social, intersubjective prerequisites, above all a language. In this process, the clear and dualistic difference between the subject and object is blurred.

Man manifests his reality in what Kosík calls objective activity. In forming himself as a socio-historical being he produces (Kosík 1976, p. 70):

- material goods, the material-sensory world based on work,
- social relations and institutions, the sum of social conditions, and
- based on these, he produces ideas, emotions, human qualities and corresponding human senses.

Thus, in objective activity human subjects create objects and social conditions which impact human consciousness and in turn create the conditions for further activity. In this activity two forms of the objective emerge. The transformation of nature into objects like tools, buildings, roads, information systems, etc. is called objectification by Kosík. The appearance of social conditions apprehended and consolidated as objects for the subjects is called objectivation. In objective activity, man cannot be separated from his objects: “The essence...”
of man is the unity of objectivity and subjectivity” (Kosík 1976, p. 70). Praxis is the realm of human being where socio-human reality is formed:

“The interconnection of objectified and objectivated praxis of mankind, labelled as substance, objective spirit, culture or civilization, [...] forms the historically attained ‘reason’ of society, which is independent of any particular individual and is thus transindividual, but which really exists only through the activity and reason of individuals. The objective social substance, in the form of materialized production forces, language and forms of thinking, is independent of the will and consciousness of individuals, but exists only through their activity, thinking and language” (Kosík 1976, p. 145-146)

3.1.2 Epistemology

The ontological position of praxis has imperative implications concerning epistemology, that is, questions about what knowledge is and how we achieve or produce reliable, well-grounded knowledge. In praxis, the dialectical relation between subject and object is regarded as the genesis of human knowledge.

The world becomes object when the subject puts herself in relation to the world that exists independently of her. This can be done only by some sign system through which the subject can express her perceptions and make them meaningful to herself and others. But any such sign system, above all language, cannot be individual, it is out of necessity intersubjective. This means that only what is intersubjective can become objective knowledge of the totality of the world.

Knowledge as we know it is not possible without an intersubjective, shared meaning about “what there is”. The acquirement of knowledge is an active construction process in which the subject influences the world, transforms it and is in turn transformed by the world. The cognitive structures acquired in this process are changed when new aspects are assimilated. “Epistemological problems are always the problems of individuals in social contexts” (Israel, 1980, p. 21). Our mental, cognitive functions are social in essence.

“This [dialectical] relation between the two basic elements – subject and object – makes praxis the central category of epistemology” (Israel, 1980, p. 92)

3.1.3 The dialectical relation

What do we mean by a dialectical relation? According to Israel (1980, p. 132) a dialectical relation is characterized by the following:

1. The elements in the relation form a unity or totality.
2. The elements are different, i.e., each element can be identified as something specific.
3. The elements depend on each other in a contradictory way.
4. The mutual dependency between the elements is not random or contingent. One element cannot be conceived of without the other.
5. The elements have something in common.
This way of understanding a relation has profound implications for how we apprehend the parts-whole relationship. Consider the following example (from Levins & Lewontin, 1985): A person cannot fly by flapping her arms, no matter how much she tries, nor can a group of people fly by all flapping their arms simultaneously. But people do fly, as a consequence of the social organization that has created aeroplanes, pilots, and fuel. It is not society that flies, however, but individuals in society, who have acquired a qualitatively new property they do not have outside society. Moreover, the parts of the aeroplane acquire new properties: they can fly by being parts of the aeroplane.

“But the ancient debate on emergence, whether indeed wholes may have properties not intrinsic to the parts, is beside the point. The fact is that the parts have properties that are characteristic of them only as they are parts of wholes; the properties come into existence in the interaction that makes the whole.” (Levins & Lewontin, 1985, p. 273)

In the dialectical process the individual / subjective reality transforms the social / objective reality and is in turn transformed by it. Giddens (1984) calls this the "duality of structure". The structures are produced by human action and at the same time, human action is made possible by and constrained by the same structures. The relationship between the individual / subjective and the social / objective is a dialectical relationship.

“[The] relationship between man, the producer, and the social world, his product, is and remains a dialectical one. That is, man [...] and his social world interact with each other. The product acts back upon the producer. Externalization and objectivation are moments in a continuing dialectical process. The third moment in this process [is] internalization by which the objectivated social world is retrojected into consciousness in the course of socialization.” (Berger & Luckmann, 1966, p. 78)

3.1.4 Everyday praxis phenomena and essence

In everyday life man is active to produce the necessary means for survival. He manipulates and attends well-known and familiar tools, systems, institutions, etc. to reach whatever goals are at hand. The circumstances in which this is carried out and the activities in doing so are called “care” and “procuring” by Kosik. Usually, the circumstances in which procuring is carried out are taken for granted and fixed.

We cannot understand the complex totality of praxis directly. What is mediated by our immediate sensory and perceptional experiences is determined in everyday life. We experience the phenomena but not the essence. For example, in our daily life we use money, a piece of paper or metal, without ever reflecting upon what the fundamental character, the essence, of money is. The essence reveals itself in moments of breakdown of everyday life, for example at wartime when hyperinflation may occur. The appearance of reality hides the essence of reality:

“Since care is the entanglement of the individual in social relations seen from the perspective of the involved subject, it also amounts to trans-subjective world seen by that subject. Care is the world in the subject. The individual is not only that which he considers himself or the world to be: he is also a part of the situation in which he plays an objective trans-individual role of which he may be quite unaware” (Kosik 1976, p. 38)
CHAPTER 3

This means that the perception is only the first step in the knowledge production process. The essence is arrived at by reflection on the immediate present and manifested in theories about the relations between fundamental categories. The phenomena become manifestations of the essence. If we want to change, for example, the way the coordination task is carried out in daily coordination work, we must first analyze the phenomena and then try to reach the essence of coordination work by producing theoretical knowledge of it.

“Familiarity is an obstacle to knowledge” (Kosík 1976, p. 46)

3.1.5 Time and space

Our daily world can be apprehended as consisting of particulars: things like chairs, books, etc. The identification of particulars is facilitated by putting them into a time-space system. This system

“... has a peculiar extension and penetration which qualifies it in a unique way as a frame for organising our thoughts regarding particulars” (Strawson, 1964, p. 25)

When man transcended the level of instinct activity for immediate needs satisfaction, the dimension of time became formative of man. It became possible to postpone immediate consumption for consumption in the future. Through labor man transforms nature and realizes human intentions in it, and in this process, man is transformed by his products. The result of labor has a duration. From being an object in transformation during labor, it ends as an enduring object which remains after man through labor has produced it. It is possible for man to think in the past, present and future.

“As objective doing, labor is a special mode of identity of time (temporality) and space (extension), as two fundamental dimensions of human being, of a specific form of man's movement in the world” (Kosík 1976, p. 122)

3.1.6 Basic assumptions about human activity

In the proceeding sections we have discussed the nature of human activity from the abstract, philosophical point of view of praxis. We have outlined an ontology and epistemology based on praxis. From this it follows that the dialectical relation is apprehended as the genesis of human knowledge and that the dimensions of essence / phenomena and orientation / temporality are fundamental characteristics of human activity. From this discussion we may conjecture a set of basic assumptions about human activity³. These assumptions should be regarded as ontological assumptions which can be questioned but not further disseminated into even more basic assumptions. This is a choice we make which reflects our praxis perspective.

The basic assumptions are:

³. This is a similar approach as the one taken in the FRISCO (A Framework of Information System Concepts) report (Verrijn-Stuart, 2001).

58
[a] There is an objective world which exists independent of, but related to human beings.
[b] The genesis of human knowledge is the interaction between humans and their environment. The knowledge and capabilities of humans are located in the individual, but they emerge in interaction with the particular situations in time and space in which the individual engages during her lifetime. The dialectical relation is the fundamental character of interaction.
[c] The interaction relates individual cognition to phenomena in its environment. The interaction is mediated by signs of various types which signify which phenomena are perceived and interpreted as relevant.
[d] Language is a sign system which emerges in interaction.
[e] Individual humans act, but not in isolation. Human actions are socially oriented. Socially organized human activity is the basis for the individual understanding of the world.
[f] In a socially organized human activity the actors have a motive for coming together, which is the reason why the activity exists. The actors are working together on an object to produce a certain outcome. The object is the main driver for organizing the activity.
[g] Human activity is contextual and situational. A context has a focus and is limited in extension. The focus may change. When the activity crosses context boundaries a translation between the structures of relevance in each context may become necessary.
[h] A context for human activity has a stabilizing core which provides a proper balance between order and disorder in the context. The core is a prerequisite for meaningful actions.
[i] There is an orientational relevance structure in a context. This structure signifies the phenomena actors perceive in the context and how these relate to each other.
[j] There is a temporal structure in a context. This structure signifies how actions in the context are coordinated.
[k] The capabilities and knowledge of actors in the context are achieved in an ongoing alteration between reflection and action. Change is regarded as endemic rather than exceptional.
[l] The use of tools is an intrinsic part of human activity.

These assumptions are the basis for the coordination constituents which will subsequently be grounded in a semiotic and cognitive analysis (see Section 3.3 Grounding the Activity Domain Theory).

3.2 Signs as mediators of the dialectical relation
In the previous sections we discussed some general features of the praxis perspective. We will now proceed to concretize this perspective. To this end we start with a general discussion of the place of signs in praxis.
3.2.1 A sign example

What is a sign? Suppose I am looking out of my kitchen window one morning and find that the Swedish flag is hoisted on the flagpole. I see a sign. A blue piece of cloth with a yellow cross on a wooden pole stands for a country with all the connotations attached to this. If the flag is hoisted all the way up on the pole it signifies feelings of happiness and solemnity. Somebody is having a birthday or is getting married. However, if the flag is hoisted only halfway up, it signifies sadness and sorrow. Somebody has died.

A sign may have a temporal aspect. For example, the flag is supposed to be hoisted in the morning and lowered in the afternoon. When a military flag is lowered at reveille, silence is imposed. A sign may also be situated both in time and space. The country signified by the Swedish flag borders onto other countries which are signified by other signs / flags. The signification of the flag also depends on the historical situation of the country it signifies. For example, the swastika was originally an old Indian symbol for the sun. After the second world war it has become a symbol for all the atrocities associated with Nazi-Germany.

The essence of the sign is that it has a physical, material side – the **signifier**, and a representative side – the **signified**. A sign stands for something outside itself. Also, the signifier and signified may change more or less independently. One and the same signifier may stand for quite different phenomena and the same phenomena may be signified by different signs.

"Signs [...] are particular, material things; and [...] any item of nature, technology or consumption can become a sign, acquiring in the process a meaning that goes beyond its given particularity. A sign does not simply exist as part of a reality – it reflects and refracts another reality [...]" (Volosinov, 1986/1929, p. 10)

3.2.2 The nature of the sign

What is the essence of a sign? How are signs formed and maintained in society? Are signs mental entities or entities outside the brain? These and other questions have aroused considerable controversy. According to de Saussure, a sign is the unity of the signifier and signified in the brain:

"[Both] terms involved in the linguistic sign are psychological and are united in the brain by associative bonds. This point must be emphasized." (de Saussure, in Innis, 1985, p. 36).

The sign is a relation in the brain between *concepts* (‘mental facts’) and *sound-image* (‘linguistic sounds’). When a word is spoken, the physical sound-wave uttered by the speaker is perceived by the auditory-sensory receptors of the hearer, recognized by the sound-image in the brain and associated with the concept that the word stands for. This means that some regular structure of connected neurons is associated with a concept.

From Pierce we have the following definition of a sign:

---

4. For a comprehensive selection of semiotic writings, see Innis, 1985
THE ACTIVITY DOMAIN THEORY

“A sign, or representamen, is something which stands to somebody for something in some respect or capacity. It addresses somebody, that is, creates in the mind of that person an equivalent sign, or perhaps a more developed sign. [...] The sign stands for something, its object.” (Pierce, in Innis, 1985, p. 5).

Thus, Pierce accentuates the outward side of the sign. The sign stands for something in the environment that means something for somebody. Pierce distinguishes between three types of signs: indices, icons and symbols (ibid., p. 9). An index is a sign where there is a connection, usually of causality, between sign and referent like smoke as an index of fire. An icon visually resembles what it represents for example, a photograph. A symbol is a sign where the association between the sign and what it represents is arbitrary like the words in a language.

Morris brings yet other dimensions to the sign concept, building on Pierce:

“Semiosis (or sign process) is regarded as a five-term relation – v, w, x, y, z – in which v sets up in w the disposition to react in a certain kind of way, x, to a certain kind of object, y (not then acting as a stimulus), under certain conditions, z. The v’s, in the cases where this relation obtains, are signs, the w’s are interpreters, the x’s are interpretants, the y’s are significations, and the z’s are the contexts in which the signs occur.” (Morris, in Innis, 1985, p. 178)

Morris emphasizes the action side of the sign as well as the context in which semiosis or sign formation takes place. The interpretant is conceived as dispositions of an organism to react according to signs. Thus, the interpretant is a mental entity in the organism, the interpreter. The signification is the apprehended properties of the object or phenomena towards which the action is directed. Signification is always context dependent.

3.2.3 Language as a system of signs

A human written or spoken language is in general considered to be the sign system par excellence, even if other sign systems also have been analyzed. One of the main controversies in language philosophy concerns language as an individual speech act (parole) versus language as a system (langue). It is obvious that both aspects exist.

In daily life, individuals are engaged in situated, fluent, more or less random verbal (and nonverbal) communication. A particular speech act is initiated by a speaker and taken up by one or several hearers, which in turn become speakers and hearers again. The words and sentences are generated by and directed to individuals. The meanings of the words may be crystal clear or diffused and so on. Language has a truly individual / subjective character.

On the other hand, language has a trans-individual character. It can be abstracted into a system of language parts like subject, verb, pronoun, etc. and structured in formal grammars. As a system, language seems to be very stable during the lifetime of an individual, yet it changes over time. Language has a truly social / objective character.

In linguistic thought, two trends can be identified where the essence of language is placed at either one of these poles. The two trends are labelled individual subjectivism and abstract...
CHAPTER 3

objectivism respectively by Vološinov (1986/1929). Individual subjectivism is based on the following principles (ibid., p. 48):

1. Language is activity, an unceasing process of creation (energia) realized in individual speech acts.
2. The laws of language creativity are the laws of individual psychology.
3. Creativity of language is meaningful creativity, analogous to creative art.
4. Language as a ready-made product (ergon), as a stable system (lexicon, grammar, phonetics), is, so to speak, the inert crust, the hardened lava of language creativity, of which linguistics makes an abstract construct in the interest of the practical teaching of language as a ready-made instrument.

The most prominent representative of this trend is Wilhelm von Humboldt (1999), who puts the emphasis on language, not as an object or product (Greek ergon) but as an activity (energeia) constantly renewed in interchanges among speakers. A modern representative for this trend is Gärdenfors (2000a) who states that the meaning of linguistic expressions is mental entities.

de Saussure is a major contributor to the second trend6: abstract objectivism, which is based on the following principles (Vološinov, 1986/1929, p. 57):

1. Language is a stable, immutable system of normatively identical linguistic forms which the individual consciousness finds ready-made and which is incontestable for that consciousness.
2. The laws of language are the specifically linguistic laws of connection between linguistic signs within a given, closed linguistic system. These laws are objective with respect to any subjective consciousness.
3. Specifically linguistic connections have nothing in common with ideological values (artistic, cognitive, or other). Language phenomena are not grounded in ideological motives. No connection of a kind natural and comprehensible to the consciousness or of an artistic kind is obtained between the word and its meaning.
4. Individual acts of speaking are, from the viewpoint of language, merely fortuitous refractions and variations or plain and simple distortions of normatively identical forms; but precisely these acts of individual discourse explain the historical changeability of linguistic forms, a changeability that in itself, from the standpoint of the lan-

5. A conspicuous example of this is the phenomena in the Swedish language of “särskrivning” (to write in two words what is one). For example: “I’m writing on my doctoral thesis” becomes in Swedish: “Jag skriver på min doktors avhandling” (I’m writing on my doctor’s thesis). The meaning of the sentence is completely altered which passes unnoticed by an increasing number of individuals.
6. Gärdenfors is actually referring to de Saussure for grounding his mentalists arguments, which may seem a bit odd. However, de Saussure was of course very conscious about the individual aspects of language. This aspect though could not be the basis for scientific analysis of language. Such an analysis had to be based on social facts, ideas which originated in Durkheim (1972).
THE ACTIVITY DOMAIN THEORY

guage system, is irrational and senseless. There is no connection, no sharing of motives, between the system of language and its history. They are alien to each other.

Vološinov rejects abstract objectivism as the actual mode of being of linguistic phenomena: “This system cannot serve as a basis for understanding and explaining linguistic facts as they really come into being” (ibid., p. 82). The problem of abstract objectivism is that it rejects the speech act as something individual. On the other hand, the problem of individual subjectivism is that the speech act becomes something confined to the individual psychic life of the speaker. Thus individual subjectivism rejects the inert characteristics of language.

The truth according to Vološinov can be found only in a dialectical synthesis of these two trends. The speech act and its product – the utterance – is a social phenomenon. The mode of being of linguistic phenomena and signs in general is intersubjective and relational:

“Every sign, as we know, is a construct between socially organized persons in the process of their interaction.” (Vološinov, 1986/1929, p. 21)

3.2.4 The sign apprehended as a relation

From the proceeding discussion it is clear that there is little consensus about the essence of a sign. We will therefore shortly indicate how we understand the sign. With de Saussure, we will assume that a sign is the unity between a signifier and a signified. For him this unity is confined to the brain. However, like Vološinov we claim that “any item of nature, technology or consumption can become a sign, acquiring in the process a meaning that goes beyond its given particularity” (Vološinov, 1986/1929, p. 10). This means that anything the cognitive system can perceive may become a sign. The signifier side of the sign relation relates something physical in the environment of an organism to ‘sensory images’ in the brain. This might be brought to the organism in the form of sound waves, photons, chemical scents, etc. For example, in Figure 14, the visual image of black dots as in “pyramid” has a corresponding pattern in the neuron network of an individual that has interacted with the environment long enough to form that pattern. This pattern is associated with a concept in the brain. The signified side of the sign relation associates this concept with some phenomena in the environment, which in turn may be apprehended as a sign.

Figure 14. Signification as a relation
Thus, a sign may be apprehended as the mediator of the dialectical relation between the cognitive system in an organism and its environment. Through the sign, the objective world becomes meaningful to the subjective organism which in turn makes the world meaningful by acting in it:

“[Consciousness] itself can arise and become a viable fact only in the material embodiment of signs [...]. Understanding is a response to a sign with signs [...]. Signs emerge, after all, only in the process of interaction between one individual consciousness and another [...]. Consciousness becomes consciousness only [...] in the process of social interaction” (Vološinov, 1986/1929, p. 11)

It is important to keep in mind that the association between signifier and signified takes place entirely in the brain. For symbols this association is arbitrary, which means that the same signifier may be associated with many signified phenomena or, contrariwise, that the same signified phenomena may be signified by many signifiers. For example, a torch is a physical object which might signify heat, light or the olympic games depending on the context in which it is apprehended. An example of the opposite is that the torch can be signified by a sound wave (somebody utters the sound ‘torch’), a written word (torch) or a picture illustrating a torch.

By apprehending the sign as a mediator between the individual and social facets of the socio-human reality the sign can be utilized both for explanatory and constructive purposes. For example, introducing a new icon in an information system means that an association between the icon / signifier and that signified by the icon must become established in the brains of the users of the information system. Problems in doing so may be analyzed in terms of similarity between the icon and what it signifies. For constructive purposes a semiotic perspective in the same situation might be to amend the icon with other signifiers (text, sound, etc.) in order to minimize the possibilities of misinterpreting the icon.

Moreover, as we saw from the example in Section 3.2.1, the sign can be associated with salient features of praxis like contextuality, temporality, orientation, etc. This makes the sign a suitable vehicle for further analysis of the dialectical relation. To this end we will discuss a stratified model of this relation in the next section.

3.2.5 A prevailing ontology of cognition - meaning emerges from the head

In order to analyze this interaction further we will elaborate on a layered model of the cognitive system given by Gärdenfors (2000a). Instead of apprehending this model as a representation of the individual cognitive system we will apprehend it as a model of the dialectical relation. This enables us to balance individual cognitive aspects with social aspects. In Section 3.3 Grounding the Activity Domain Theory this will be utilized in to ground the coordination constituents.

7. It may be argued against our understanding of the sign that the phenomena of inner speech or signification of phenomena that does not exist (such as e.g. fairies or trolls) is not considered. However, we still regard the interaction as basic for the emergence of this type of signification. It is not possible to conceptualize a fairy without interaction with other individuals.
We will use Gärdenfors “Conceptual Spaces – the geometry of thought” (Gärdenfors, 2000a) as a point of departure. The ontology and epistemology that underlies this theory is that social meaning emerges from the heads of individuals:

“The core idea is that meanings of linguistic expressions are mental entities – meanings are elements of the cognitive structure in the heads of the language users.” (ibid., p. 154)

Clearly, this position differs from ours where the dialectical relation is regarded as the gene-

esis of all cognitive effects. Gärdenfors’ position runs into problems regarding semiosis (the formation of signs) and the question of how the mental entities came into the head to begin with. Another problem is the social nature of meaning: how are shared meanings among individuals possible? Gärdenfors realizes this difficulty:

“A fundamental assumption of this analysis, however, has been that the conceptual structure belongs to some individual language user: the meanings of words reside in the heads of individuals. On the other hand, it is also obvious that language is a social phenomenon. So how can individual conceptual structures become social? If each person can mandate his own cognitive meaning, how can we then talk about the meaning of an expression? [...] how can it be established that we talk about the same things? [...] the communicative question is a genuine problem for cognitivist theories of semantics.” (ibid., p. 189-190)

The way Gärdenfors tackles this problem is to start from the observation that any association between a thing and a concept must take place in the brain. There is nothing in the external world that can hold this association. Therefore, in a communicative situation where a speaker and hearer are communicating about things, their cognitive systems must be aligned to some degree in that particular situation. This imposes certain constraints on the individual cognitive representations. These constraints are according to Gärdenfors related to the linguistic power structure:

“I argue that the social meanings of the expressions of a language are indeed determined from their individual meanings – the meanings the expressions have for the individuals, together with the structure of linguistic power that exists in community. The linguistic powers concern who is the master of meaning, who decides on what is the correct meaning of an expression in a society”

(ibid., p. 197)

One assumption is that the situation will drive the representations towards the most econom-

ical way of achieving a reasonable shared meaning. The social interactions will result in a set of communicable references which are found in the cognitive structures of both the speaker and the hearer. This stance leads to a chicken and egg problem: are the cognitive structures prerequisites for successful communication or are they emergent results of successful communication? The answer given by Gärdenfors is “both”:

“My sociocognitive position can be summarized as follows: meanings are not in the head of a single individual, but they emerge from the conceptual structures in the heads of the language users together with the linguistic power structure.” (ibid., p. 202)

8. Some branches of cognitive science have started to move away from this position (see for example Rogers & Ellis (1994) and Wertsch (1991).
CHAPTER 3

This position is in essence not far from our praxis perspective, although ontologically and epistemologically different. On the other hand, the praxis perspective is quite compatible with the position that meanings are elements of the cognitive system of humans. We agree that the association takes place in the brain only. However, the formation of the association is through the dialectical relation with the environment. Interaction and intersubjectivity are the generating mechanisms of both individual cognitive structures and the social meaning of language:

“By its very existential nature, the subjective psyche is to be localized somewhere between the organism and the outside world, on the borderline separating these two spheres of reality. [...] the organism and the outside world meet here in the sign” (Vološinov, 1986/1929, p. 26)

The praxis perspective provides an answer to the problem of semiosis and genesis of mental entities without referring to reified agents like ‘the linguistic power structure’. Altogether though, the Gärdenfors’ theory provides a well-grounded point of departure for our analysis, provided that we view his theory from our praxis perspective.

3.2.6 Shifting the focus - from heads to interactions

Gärdenfors’ knowledge interest is to model representations of the cognitive system in organisms for explanatory and constructive purposes such as for example developing artificial neuron networks. According to him there are two dominating approaches for modelling representations of the cognitive system: the symbolic and the associative. A particular type of association is connectionism where the information carrying elements are modelled after the neural network in the brain.

This has resulted in two branches of research concentrating on either the symbolic or connectional approaches. In the symbolic approach the brain is regarded as a Turing machine which manipulates symbols, and in the connectional approach the brain is regarded as a system of connected neurons which respond to stimuli of various kinds. These approaches should be seen as complementary rather than competing since the symbolical representation presumes the connectional representation in the sense that the nervous system is a prerequisite for symbol formation.

However, Gärdenfors claims that important aspects of the cognitive system such as concept acquisition, i.e., how concepts are learnt and internalized, cannot be adequately captured by either of these approaches. It is also unclear how to model the transition between the connectional and the symbolical levels since the formalisms used are very dissimilar. The step from neuron models to symbols becomes too large. To this end, Gärdenfors proposes a third,
THE ACTIVITY DOMAIN THEORY

intermediate level: the conceptual one. Thus Gärdenfors winds up with three levels: the symbolical, conceptual and connectional ones (see Figure 15):

![Layered model for cognitive systems](image)

Figure 15. A layered model for cognitive systems (adapted after Gärdenfors, 2000a)

Since we want to shift the focus from the individual to the relational, we will proceed from the Gärdenfors model. However, rather that viewing the model as a model of the cognitive system of individual organisms, we will apprehend it as a model of the dialectical relation between an organism and its environment. Gärdenfors’ main interest is focused on the conceptual level as an intermediate level between the connectional and symbolic ones. We will retain the connectional and conceptual levels from Gärdenfors. However, the symbolical level in Gärdenfors refers, as we saw, to symbol manipulation according to some rules. This representation comes from Artificial Intelligence (AI). Since we are not interested in this kind of representation we will conceive the symbolic level as a linguistic level where the written and spoken word is the main model construct. Thus, symbols will be included in the linguistic level.

The notion of ‘levels’ suggests that there is an inherent, hierarchical ordering between the levels. The biological, neural apparatus of a particular organism at a particular instant in time may appear to be more ‘basic’ than the concept or symbol formation that takes place during the organisms’ life time. However, without interaction the cognitive, genetically inherited system of an organism would not have evolved in the first place. When placing the dialectical relation in focus, the evolution in the different layers becomes rather a matter of long-term or short-term interaction. Since we want to stress the formative aspects of the model, we will replace ‘level’ with ‘stratum’, which has connotations of history, sedimentation, diffusion, etc.

9. A comprehensive set of arguments for moving towards a position like this is given by Gedenryd (1998) where he proposes a model called interactive cognition, which bare many similarities with the one proposed in this study: “... interaction realizes the link between mind and environment in the cognitive process” Gedenryd (ibid, p. 12).

10. In Lemke (2000) a distinction is made between a ‘topological’ and ‘typological’ levels. These corresponds roughly to the connectional and linguistic levels. However, like Gärdenfors we believe that there is a need for the intermediate, conceptual level.
CHAPTER 3

In summary, our model can be illustrated as in Figure 16.

![Figure 16. A stratified interaction model](image)

As with any model, this one is a theoretical construct with a particular purpose in mind, in our case to articulate the interaction for operational purposes in connection with the coordination of the development of complex systems. The model should not be apprehended as a representation of ‘real’ interactions in nature. There is nothing in the external world which materializes the dialectical relation. The cognitive apparatus of organisms still resides in the nervous system in the brain. Likewise, it does not change the fact that there exists a natural and social world. The point of focusing on the dialectical relation rather than on either the individual or the social, is that we consider this to be a more viable way to address problems regarding semiosis and genesis, i.e., formative and creative aspects of the socio-human world. Thus, our scientific position w.r.t. this model is basically instrumental (see e.g. Chalmers, 1976).

3.2.7 Biological prerequisites

The biological prerequisites for the model are those which enable an organism to interact with its environment through its cognitive system. Even on a very basic level we find interactions as a driving force. For example, an alien substance from the environment may trigger a cancerous reaction the cell dividing process. This may activate so called T-cells (Cytotoxic T lymphocytes) which recognize surface markers (antigens) on other cells in the body that label those cells for destruction. The character of the interaction in this case would be the structures of the proteins of the antigen and T-cell, which make it possible for the T-cells to adhere to the cell marked with the antigen.

One element of this base is the sensorimotorical system. Sensations are immediate *sensory impressions* like tastes, scents, colours, tickles, pain, etc. The biological role of sensations is to direct the organism’s attention to what is happening to it at this very moment. Sensory receptors provide information to the organisms’ nervous system where it is processed in order to take appropriate action. Even very simple organisms have sensations. However, this
might not necessarily lead to learning. Very complicated, genetically programmed patterns of actions may be carried out without any learning taken place. The character of the interaction depends on the type of sensation in the interface between the sensory receptor and the environment. External cues or signals are converted or transduced into information that the organism uses to survive. Gärdenfors notes that the sign type ‘index’ corresponds to signals, i.e., stimuli closely related to events in the immediate spatial and temporal environment of the organism (Gärdenfors, 2000b, p. 141).

Another element of the biological base is perceptions. The role of perceptions is to inform the organism of what is about to happen in the environment, not only what is happening right now which the sensations inform about. According to Gärdenfors (2000b), perceptions for advanced organisms are associated with a type of simulators in the brain, by which different types of actions can be anticipated before they are carried out\(^\text{11}\). The information from the sensory receptors is fed into the simulator and amended with other signals which might emerge from stored representations rather than sensory receptors. An interpretation of the situation is carried out and the relevant action is performed. Simulators presume a learning capability, which means that more complex nervous systems are involved than for pure sensations. However, the perceptual simulators are always related to sensory impressions.

3.2.8 The connectional stratum

The sign-related elements in the connectional stratum are a large number of simple but highly connected neurons. These receive excitatory or inhibitory inputs and transmit activity to other units according to some function of the inputs. The processing is done in parallel and the strength of the connections between the units changes according to made experiences from the interaction with the environment. The result of the organism’s interaction with the environment is the formation of patterns of connected neurons. The high dimensionality of the connectivist stratum makes it implausible that the strength between connections is independent since the learning curve would become prohibitive long. It is therefore likely that the connection strengths are correlated according to high-level structures in the conceptual strata with a substantially lower dimensionality.

This raises the interesting issue of where the control of the brain functions lies. The neural patterns and thus the control of its formation, cannot be explained in terms of single neurons in the connectional stratum. Rather, the control is an emergent phenomenon which appears in the interaction with the environment through the higher strata (see e.g. Sperry, 1976). The formation of the control and our consciousness is a two-way movement: an upward causation from the connectional stratum and a downward causation from the environment and higher strata (see e.g. Lemke, 2000). Thus, the formation of the connectional stratum is dialectical in character since the connectional stratum provides means of apprehending and possibly changing the environment and at the same time is formed by that very environment.

\(^{11}\) For example, a cat can ‘simulate’ a mouse behind a hole in the wall and sit waiting for hours for it. A snake does not have this capability. As soon as a pray has disappeared out of the snake’s sensory range, the attention of the snake ceases. (Gärdenfors, 2000b, p. 45)
3.2.9 The conceptual stratum

Human beings have a cognitive capacity for generating ordering relations of stimuli based on what Gärdenfors calls *quality dimensions* (Gärdenfors, 2000a, p. 6). Examples of such quality dimensions are temperature, weight and spatial dimensions like height, width and depth. Some quality dimensions are culturally dependent, e.g. time which may be linear or circular (ibid., p. 28).

The quality dimensions are associated with geometrical, topological or ordering structures (time being one of them) and they are the basis for building what Gärdenfors calls *conceptual spaces*. Concepts are vital to similarity judgements and the structure of quality dimensions of a conceptual space will make it possible to talk about distances along the dimensions. Distances in the conceptual space are closely related to similarity judgments, hence the subtitle of Gärdenfors work (“The geometry of thought”).

Thus, the sign-related elements in the conceptual stratum are *concepts* which are spatially related to each other (ibid., p. 5). In this way, the many dimensions of neural network converge into regular structures which can be conceived of in the conceptual stratum through a reduction of neural dimensions to quality dimensions (ibid., p. 240).

The quality dimensions are independent of symbolic representations and they are more fundamental (ibid., p. 43). Thus, the conceptual stratum in turn is a basis for the uppermost, linguistic stratum. The conceptual stratum provides an appropriate mechanism for grounding the linguistic stratum which would otherwise have to be grounded directly in the connectional stratum. A major point in Gärdenfors’ contribution is that such a direct, short cut grounding cannot be convincingly argued for:

“*[The] symbolic and connectionist representations are not sufficient for the aims of cognitive sciences; many problems are best handled by using geometrical structures on the conceptual level*”

(Gärdenfors, 2000a, p. 3)

3.2.10 The linguistic stratum

The relation between language and semiotics (the study of signs) is a problematic one. In general, language is considered to be the most important sign system (de Saussure, in Innis, 1985). However, as we saw in Section 3.2.3 *Language as a system of signs* the nature of language itself is ambiguous. For example, Benveniste claims that language is the interpreting system of all other semiotic system (in Innis, 1985, p. 239).

For our purposes we will understand linguistics as the study of human speech including the units, nature, structure, and modification of language. Some sub-fields in this area are phonology and phonetics (sounds of language), semantics (meaning), syntax (grammar), etc.

The sign-related elements are symbols which build up words or elements of words (letters or syllables). These are communicated as strings of sounds or texts, one after another. The relationship between the signifier / symbol and the signified is mainly arbitrary in the sense that the same signified, for example a ‘tree’ has different signifiers in different languages.
In the stratified model the linguistic stratum presumes the conceptual stratum. Language is dependent on a pre-conception. This issue has been addressed by the branch of semiotic theory called cognitive semantics. The gist of this theory is to regard semantics as a relation between linguistic expressions and a cognitive structure. Meaning is a conceptual structure, not a logical truth condition. The semantic elements are constructed from geometrical or topological structures. Gärdenfors (1992) describes the use of so-called image-schemas to represent perception, memory, and semantics. These image-schemas have an inherent spatial structure and are supposed to be represented in the cognitive structure of the individual. For example, the meaning of “over” and “under” can be represented by the image-schemas given in Figure 17:

![Image-schema for "over" and "under" (adapted after Gärdenfors, 1992, p. 99)](image)

“Over” is apprehended as a relation between two objects. One object is in focus: the trajector (marked with a bold periphery). The other object is called the landmark. The only difference between “over” and “under” is that the focus changes. The image-schema shows that there is a certain relation between two objects which is valid during a certain time period.

To summarize: we have interpreted the Gärdenfors’ layered model representing the cognitive system of individual humans as a model of the dialectical relation between the human individual and her environment. In this model signs are the mediators between the individual and the environment. The different strata (connectional, conceptual, and linguistic) represent a gradual focal shift from the biological realm to the social realm in the formation of socio-human activity. As Volosinov we consider the social realm to be the dominant one:

In the lower strata of behavioural ideology, the biological-biographical factor does, of course, play a crucial role; but its importance constantly diminishes as the utterance penetrates more deeply into an ideological system. Consequently, while biographical explanations are of some value in the

12. An interesting observation is that the conceptual representations of Gärdenfors (concepts, properties, domains, objects, regions, relations, etc.) are similar to the constructs of object-oriented relational models (types, type hierarchies, attributes, objects, relations, etc.). This implies that an object-oriented way of modelling praxis may be well aligned with the conceptual spaces in our brains. Thus, the achievement of intersubjectivity about the coordination praxis may be facilitated by choosing an object-oriented modelling notation. We will return to this issue when we discuss the construction of the Framework in chapter Chapter 5 The structure of the Framework.
lower strata of experience and expression (utterance), their role in the upper strata is extremely modest. (Vološinov, 1986/1929, p. 93)

3.3 Grounding the Activity Domain Theory

In the proceeding sections we have discussed a stratified model of the dialectical relation in which signs mediate between the individual / subjective and social / objective aspects of praxis. Furthermore, from the praxis philosophy we have extracted a set of basic assumptions about human activity.

In this section we will propose a set of coordination constituents which we conjecture is fundamental for coordinating socio-human activity. We will then analyze each one of these constituents with respect to the strata in the model of the dialectical relation. If the same constituent appears in several of the strata we argue that this constituent is indeed fundamental for the coordination of human activity. Thus, any such constituent should inform the design of the Framework. Together with some notions from Activity Theory (motive, object and outcome) these constituents will be congregated into the construct of activity domains which is the core of the Activity Domain Theory (see Section 3.4 The Activity Domain Theory – structured praxis).

3.3.1 Coordination constituents - a conjecture

From the basic assumptions given in Section 3.1.6 we conjecture that the following elements are fundamental for the coordination of socio-human activity:

- **Intersubjectivity**: This is a prerequisite for a consorted action in a context. Shared meaning arises in the interaction between humans and is mediated by various sign systems, above all language.

- **Contextuality**: Humans have an inherent capability to apprehend contexts and change focus between different contexts. When the focus is changed spatial and temporal structures are reorganized in the sense that new phenomena and action patterns become attended to while other diminish below the attention horizon.

- **Domain transition**: The practice of humans takes place in a unreflected, commonsense understanding of the world where phenomena are taken for granted. In order to understand the essence of the phenomena these must be revealed, reinterpreted and translated in order to be comprehensible. In order to do so a transition must take place between the to domains of discourse.

- **Experiential learning**: Human knowledge and capabilities are acquired in an ongoing iteration between reflection and action.

- **Orientation**: Humans have an inherent capability to structure the world spatially in a context. The orientation is achieved by classifying and categorizing phenomena in a context and how they are related to each other. The spatial structure is static in the sense that phenomena and their relations change only slowly as experiences are accumulated by the individual.

- **Temporality**: Humans have an inherent capability to structure the world temporally which relates to the order in which events occur in a context and how the individual
reacts to and impacts these events. The temporal structure is interrelated to the spatial structure.

- **Stabilizing core**: Stabilizing structures are necessary in human activity. A stabilizing core is a prerequisite for actions, and provides a proper balance between order and disorder.

- **Tool usage**: The making and use of tools are inherent in human activity.

- **Motive, outcome and object**: Human activity has a motive which is the reason why the activity exists. In the activity actors work on object(s) in order to produce an outcome.

In the following sections we will discuss these in turn.

### 3.3.2 Intersubjectivity and shared meaning

“A word is a bridge thrown between myself and another” (Vološinov, 1986/1929, p. 11)

Our point of departure in praxis means that we regard the dialectical relation as the genesis of human knowledge. Like Israel (Israel, 1980, p. 30 ff.) we believe that the use of language presumes an intersubjective understanding. To speak means to share. Language is learned in interaction with others and is an integrated part of reality in the sense that it constitutes reality.

Regarding the cognitive aspects of intersubjectivity Gärdenfors quotes Freyd:

> "For example, Freyd (1983, 193-194) puts forward the intriguing proposal that conceptual spaces may evolve as a representational form in a community just because people have to share knowledge [...] ‘But it might be that the structural properties of the knowledge domain came about because such structural properties provide for the most efficient sharing of concepts. That is, we cannot be sure that the regularities tell us anything about how the brain can represent things, or even ‘prefer’ to, if it didn’t have to share concepts with other brains.’’" (Gärdenfors, 2000a, p. 29)

The ontological position following from this statement is that meanings are in the heads. The particular meaning in one head comes about by sharing it with another head. The intersubjectivity is an optimization problem where the optimal solution is the one where all heads in a particular context share their conceptual spaces. Although we do not share this ontology, it is evident that the position put forward in the quotation indicates that the individual cognitive system is formed by intersubjectivity.

Regarding the linguistic aspect of the dialectical relation we may notice that intersubjectivity plays an important part in the “language game” concept of the later Wittgenstein (in Svensson, 1978). In this game, the rules must be encompassed by all ‘players’ or language users. A player who doesn’t follow the rules, except for occasional mistakes, cannot participate in the game. Moreover, the rules are in essence intersubjective since they must be followed by anyone participating in the game. An important corollary of this is that there must at least be two players in a language game. A single language game player is an anomaly13.

Intersubjectivity is a prerequisite for the emergence of shared meaning in a context. In current system development practices however, the importance of this characteristic is mostly
downplayed. One reason may be that shared meaning is difficult to measure and analyze, and consequently, the cost of achieving shared meaning is frequently overlooked. Another reason may be that the articulation of symbols in the semiotic layer is difficult to achieve since the relation between the signifier and signified is arbitrary. However, the lack of shared meaning may cause failures in product development (Hoopes & Postrell 1999). Thus, it is of outmost importance that the Framework supports the achievement of intersubjectivity about the coordination praxis.

3.3.3 Contextuality

Human action is situated. We act according to what comes into the focus of our attention. Memories of earlier experiences are compared with the situation at hand and possible actions are evaluated. Sometimes the situation is completely new but in most cases, the current situation is similar to situations in the past to which we can relate. Thus in a certain focal situation, capabilities relevant for that particular situation are activated, no matter in which strata in the interaction model. When we put our hand on a hot plate on a stove, the reaction to pull back the hand starts in the sensory organs. The experience is transferred to the brain where the conceptual space concerning plates on stoves is modified in order to make it possible to act differently when the same situation comes into focus the next time.

Gärdenfors claims that one aspect of the conceptual stratum is its contextual dependency: "...the salience given to various aspects of a concept may vary depending on the context...the first problem when representing a concept is to decide which are the relevant domains" (Gärdenfors, 2000a, p. 102). ‘Domains’ in the terminology of Gärdenfors’ can be understood as characteristics of concepts. For example, an apple may be characterized by the domains color, shape, texture and taste. However, the association of domains with concepts is context dependent. Gärdenfors gives the example of a piano. For someone moving the piano, the weight domain is probably a salient domain, but for a recital performer, the pitch and timbre of the tones are more likely salient domains. Thus, what is a core characteristic in one context may become a marginal characteristic in another context.

We also find several examples of contextuality as a major feature of language formation. For example, the central concepts in the language view of Wittgenstein are language games, utterance, meaning, action, language rules and context. The concrete situation in which the utterance is uttered is where the meaning of the utterance must be sought. For Wittgenstein, the context is not just the linguistic context, i.e., the relationships between the linguistic elements in a sentence, but everything which is relevant in the situation such as, for example, who utters the utterance and the body language of the speaker.

An extreme proponent of contextuality is Vološinov:

13. It may be noted that this applies also to a person ‘playing’ the game of solitaire if we consider her interaction with the cards as a game. The game only makes sense if the rules are followed.

14. Gärdénfors notes that the symbolic paradigm in cognitive sciences has by and large been a failure due to several reasons, one of them being the lack of notion of contextuality.
“The meaning of a word is determined entirely by its context. In fact, there are as many meanings of a word as there are contexts of its usage.” (Vološinov, 1986/1929, p. 79)

For Vološinov, verbal communication can never be understood and explained outside a connection with a concrete situation. None of our descriptions of the world is total, and new aspects can always be discovered. The way we describe something is not determined by inherent qualities of the things but of how we relate to them and in what context. Blue means quite different things to a physicist and an artist.

From the system development practice we know that a complex task needs to be divided into contexts like software, hardware, mechanics and radio development areas in a mobile telecommunication system. Large projects need to be divided into smaller contexts to make them manageable at all.

The notion of ‘functionality’ is important in system development. Telecommunication systems are developed and tested according to functional specifications. Usually functionality is considered to be an inherent property of the system. However, we claim that functionality is inseparably related to context. For example, the intended function of a chair is undoubtedly its ability to provide something to sit on15. In the context of sitting, some properties of the chair will be salient such as e.g. softness and sitting area. However, in other contexts, such as for example using the chair to step on to reach something, the intended function of the chair is irrelevant and other characteristics become important, e.g. the height of the chair.

There are two aspects of contextuality that have to be considered. The first one is how to characterize a particular context and the other one how to characterize context changes. We assume that some parts of the conceptual space in the brain can be articulated and made explicit in the form of signs. For example, if we want to characterize a certain development

15. In J. J. Gibson's theory of perception, human relates to the work through what he calls affordances which stands for the 'invitational' quality of a percept or an event. Thus, a part of the affordance of a hammer is its graspability, of a chair its sit-on-ability. In a sense, affordance refers to the intrinsic properties of items and events (The Penguin Dictionary of Psychology, Arthur S. Reber, 1995). It arises from the joint functioning of the organism and the individual.
situation regarding the development of a telephone switch, we might illustrate this as in Figure 18:

![Figure 18. Illustration of a focal situation](image)

Here, the complete illustration may be seen as a complex sign which should signify more or less the same phenomena for the actors involved in this particular context. This implies that some parts of the conceptual spaces in the heads of the actors correspond to the concepts, relations, properties, etc. found in the illustration. In fact, we may conjecture that whatever structure the particular part of the conceptual space may have, it can be always be transformed to the structure of the signifier associated with the conceptual space, in this case the illustration.

Thus, rather than defining a sign as an association between a concept and sound pattern as de Saussure did, we define a sign as an association in the brain between a signifier context and a conceptual context in the conceptual strata. This has the important consequence that the illustration / sign in a sense is both a map of the individual conceptual space in the brain and a map of certain objective conditions in a context. Since this map emerges in interaction, it will also to some extend be a map of the intersubjectivity among the actors. Thus, in the signifying process (the semiosis) when the contents and form of the illustration are discussed among the actors, the intersubjectivity and the signifying structure emerge dialectically and simultaneously.

We will denote the ensemble of stratified capabilities activated in a certain focal situation situated knowledge. This knowledge may be separated into knowledge concerning the concept in focus and knowledge of related concepts. These are called focal knowledge and background knowledge respectively. Other capabilities which are not relevant for a particular situation we call latent knowledge. Since these are not active we may think about them

---

16. This discussion is related to the concept of tacit knowledge coined by Polanyi (1966). For Polanyi, tacit knowledge equals background knowledge in a particular focal situation. It seems more appropriate to call this background knowledge for latent knowledge and reserve the term tacit knowledge for the connectional strata in the stratified model. For a discussion on these matters, see Braf (2000, p. 86 ff.).
as below a horizon; they are not visible. Situated capability is then bounded by a situational horizon which delimits the particular situation against other situations (see Figure 19):

3.3.4 Domain transition

A change of context will bring other concepts, properties and relations visible above the situational horizon. Focal knowledge may turn into background knowledge and vice versa. For example, a focus on the production and manufacturing of the switch in the previous section rather than on the development of the switch will make other situational capabilities more relevant. Some of the needed capabilities may be found in quite different types of praxis. In general, the capabilities in one praxis may need some capabilities from another praxis without including it as a whole. In order to be understandable, the included form of praxis must be transformed into familiar knowledge structures.

Thus, there is a need for a transformation mechanism between different forms of praxis. For example, the praxis of developing software for the switch needs some information from the praxis of hardware development regarding the signals that pass between the interface between the software modules and hardware modules. This information must be translated into a form suitable for software development, for example, binary signals rather than voltages and currents as in the hardware development praxis.

Some important aspects of contextuality and domain transition related to coordination are: How should the context be structured? What knowledge is needed in terms of focal knowledge and background knowledge? Where is the situational horizon for the coordination context located? Are there any focal changes in the coordination context? What about transformations between the coordination praxis and other forms of praxis?, etc. This implies that the Framework must at least support the notion of contextuality, focal changes and the translation between contexts.

Figure 19. Some notions regarding contexts
“Every stage in the development of a society has its own special and restricted circle of items which alone have access to that society’s attention and which are endowed with evaluative accentuation by that attention. Only items within that circle will achieve sign formation and become objects in semiotic communication” (Voloshinov, 1986/1929, p. 21-22).

3.3.5 Experiential learning

Having established that intersubjectivity is a result of interaction, we must be more precise about the process of achieving intersubjectivity. There is ample evidence that capabilities in general of higher organisms come about in an evolutionary and experiential way. The theoretical foundation of experiential learning was laid by Dewey (pragmatism), Lewin (Gestalt psychology) and Piaget (cognitive development processes)\(^\text{17}\). For example, for Piaget, intelligence and the ability to manipulate symbols result from interacting with the concrete environment. Learning is conceived of as a process in which concepts are derived and continuously modified by experience.

Moreover, acquirement of capabilities seems to take place through an alteration between modes of reflection and action. Paulo Freire defined praxis as “reflection and action upon the world in order to transform it” (Freire, 1996, p. 33).

“Within the word we find two dimensions, reflection and action, in such radical interaction that if one is sacrificed – even in part – the other immediately suffers. There is no true word that is not at the same time a praxis. Thus, to speak a true word is to transform the world.” (Freire, 1996, p. 68)

The word has an inseparable dual character of action and reflection:

“When a word is deprived of its dimension of action, reflection automatically suffers as well; and the word is changed into idle chatter, into verbalism, into an alienated and alienating “blah.” [...] On the other hand, if action is emphasized exclusively, to the detriment of reflection, the word is converted into activism. The latter – action for action’s sake – negates the true praxis and makes dialogue impossible” (Freire, ibid., p. 69).

Kolb (1984) elaborates the reflection-action modes into a process of experiential learning consisting of a four-stage cycle involving four adaptive learning modes: concrete experience, reflective observation, abstract conceptualization, and active experimentation. This implies that learners must be involved in new experiments which they can reflect on and observe from many perspectives. They must also create concepts and categories that integrate the observations into a coherent theory, which in turn may be used to carry out new experiences.

Gärdenfors shares the view of interaction as the basis of learning. The structure of concepts is established and maintained through acting in the world: “Via successful and less successful interactions with the world, the conceptual structure of an individual will adapt to the structure of reality” (Gärdenfors, 2000a, p. 156).

\(^{17}\) For a comprehensive overview of experiential learning theories, see Kolb (1984)
From the linguistic or social point of view, there are several examples of experiential learning approaches in organizational learning theories. For example, Nonaka & Takeuchi (1995) propose a model for transforming tacit knowledge in an organization into explicit knowledge and vice versa. The Deming cycle of plan-do-check-act in total quality management is another example. Kim (1993) describes a learning cycle consisting of the phases observe-assess-design-implement where people in organizations experience concrete events and actively observe what is going on. Their experiences are assessed (consciously or subconsciously) by reflecting on their observations. An abstract concept that seems to be an appropriate response is designed and tested by implementing it in the concrete world. Based on new concrete experiences the cycle is started all over again.

In system design experiential learning can be found in the concept of “daily build”, where software code is developed and tested in small development cycles. Gilb (1988) has proposed an evolutionary programming methodology. Earl (1996) has reported that an evolutionary approach seems to be superior to other approaches when implementing an information system strategy in an organization. Recently Kristensen (2001) described an evolutionary information system development methodology.

“Learning is the process whereby knowledge is created through the transformation of experience” (Kolb, 1984)

3.3.6 Orientation and temporality

In our daily life we have a strong sense of the spatial and temporal structures. Time is experienced as something qualitatively different from space. Regarding the cognitive aspect, the conceptual spaces are static in the sense that they only describe spatial relationships between concepts. However, there is obviously a temporal aspect of our cognitive apparatus since we act on orientational information derived from a focal situation. This also applies to the connectivist stratum. Greenfield (1998, p. 217) quotes the neurophysiologist Ad Aertson:

“[We] should distinguish between structural and anatomical connections on the one hand and functional or effective connectivity on the other. The former can be described as quasi-stationary, whereas the latter may be highly dynamic [...]. It appears that dynamic co-operativity is an emergent property of neuronal assembly organisation in the brain which could not be inferred from single neuron observation”18

Our cognitive apparatus seems to be equipped to recognize orientational and temporal patterns in our apprehension of the world. Gärdenfors quotes Langacker (1987):

“It is however necessary to posit a number of “basic domains” that is, cognitively irreducible representational spaces or fields of conceptual potential. Among these basic domains are the experience of time and our capacity for dealing with two and three-dimensional spatial configurations.” (Langacker, 1987, p. 5, quoted in Gärdenfors, 2000a, p. 161)

18. It can be noted that this quotation is in line with the stance of dialectical interaction as a fundamental epistemological category. Dynamic cooperation brings out new qualities even at the connectivist stratum.
CHAPTER 3

The conceptual spaces according to Gärdenfors concern static patterns only; little is mentioned about dynamic patterns or processes:

"Conceptual spaces are static in the sense that they only describe the structure of representations. A full model of cognitive mechanisms not only includes the representational form, but also a description of the processes operating on the representations." (Gärdenfors, 2000a, p. 31)

From the social point of view we also find orientational and temporal patterns. Temporal structures are to some extent culturally determined. In some cultures, time is regarded as circular rather than linear. Moreover, as has been noted earlier in Section 3.1.5 Time and space, when man could suppress instinctive action for immediate needs satisfaction, the dimension of time became a formant of man. From the organizational learning disciplines we find for example the terms conceptual and operational learning (Kim, 1993). Another example is the separation of a speech act into the locutionary (the proposition / information component) and the illocutionary (the action component). In system development we find orientational patterns like product structures, functional structures, object-relational models, etc. Temporal patterns may be found in process models, state diagrams, etc.

Orientation may be apprehended in a transferred sense as orientation between concepts which may be abstract. For coordination purposes the Framework must support the evolution of both the orientational and temporal structures of the coordination praxis. The orientational structure concerns concepts, their characteristics and their relations to each other. The temporal structure concerns the order in which activities are carried out.

"The world of everyday life is structured both spatially and temporally. [...] Temporality is an intrinsic property of consciousness. The stream of consciousness is always ordered temporally." (Berger & Luckmann, 1966, p. 40)

3.3.7 Stabilizing core

In social situations a balance between order and disorder must be maintained:

"Every social situation lives in order and disorder; every social situation holds a moment of order and a moment of disorder. But at total disorder, there is no social life and at total order there is just a social petrification. Man lives his life between the impossible possibilities chaos and petrification, but in an endless number of ways. The question is then what composition, combination or mixtum compositum is the happy one?" (Nilson, 1976, p. 10)

This implies that there is some kind of stabilizing core in a situation or context. There may be biological rules or restrictions related to natural laws, which result in death or serious diseases if violated, for example, consuming lethal food. For humans, rules may be social and codified in religious or juridical laws. Other rules are learnt in social interaction and internalized without articulation. Violating the rules in the language game of Wittgenstein means that an actor either does not understand the rule or deliberately chooses not to participate in the game. Traditions and culture are necessary to keep a society together.

Rules may be more or less comprehensive in a particular situation or context. A completely rule-free situation approaches a state of chaos or disorder, while a completely rule-control-
THE ACTIVITY DOMAIN THEORY

led situation is approaching a state of petrification. At either endpoints consorted action is impossible. Somewhere in between there is an optimal rule coverage.

In the biological realm we will find other types of stabilizing cores. For example, the DNA molecule is the basic building block in all living organisms. Although its basic structure is simple (there are only four types of nitrogenous base in the DNA molecule), it gives rise to an overwhelming abundance of organisms. Very simple rules like “do what your neighbour does” may lead to a variety of collective behaviours of social insects like honey bees or army ants.

Greenfield (1998) describes how a group of neurons with relative long-lasting connections between them, may engage neuronal assemblies when a particular stimulus is reaching the brain. Depending on the size of the neuronal assembly different forms of consciousness may arise. Small neuronal assemblies may result in a kind of consciousness that would just be reactive to whatever crossed one’s path without a lot of reflection and memories and thoughts. This form of consciousness appears in a child’s perspective and a schizophrenic person. On the other hand abnormally large neuronal assemblies would have the opposite effect. The outside world would appear remote, grey, and distant. This form of consciousness can be found in clinical depression. Thus it appears that in a developed and healthy brain, a balance between the two extremes of chaos and petrification exist.

In the social realm we can see the balance between order and disorder in many areas. For example in an organization there is a contradiction between continuity and change. If the organization is too laden with respect for old traditions it may be slow to implement necessary changes. On the other hand, if the organization changes too quickly, the gradual building of human and material resources may be impeded. Recently Sawhney and Prandelli proposed a new governance mechanism for managing distributed innovation called a “community of creation” with the purpose of blending the benefits of hierarchies and markets by offering a compromise between too much structure and complete chaos (Sawhney & Prandelli, 2001).

Another example comes from Lakatos’ anticipation of science as research programmes (see Chalmers, 1976 for a comprehensive description). In order for scientists to fully explore a certain scientific line of research, they cannot constantly be engaged in debates regarding the basic assumptions and methods of a research programme. Lakatos calls these basic assumptions the hard core of the programme. These are guarded from falsification by a protective belt of auxiliary hypotheses. As long as the hard core can be protected, the research programme can be explored to its full potential. The moment that the hard core cannot be sustained, the research programme has to be abandoned. The hard core can be apprehended as a stabilizing core consisting of fundamental constituents of the scientific community.

In a large and distributed organization like Ericsson, there will always be a need to adapt enterprise strategies to specific circumstances at the local design centres around the world. At the same time, there must exist common policies for enterprise tasks, such as, for example, how to identify products and documents. The balance between strict control and decentralization is often difficult to maintain, and the organization tends to oscillate between these extreme points from time to time. In either case, the ability to adapt to changes will be low, as illustrated in Figure 20. Thus there is a need for an overall stabilizing core in the organi-
zation. A part of this core could, for example, be common rules for identifying products and documents\(^{19}\).

The order-disorder balance is also salient in quite another area: stochastic optimization. For many years I worked with a stochastic optimization algorithm which was used extensively in the Ericsson development practice to optimize the performance of telecommunication systems. For example, the filtering function of a transmission system could be optimized by randomly adjusting filter parameters according to the algorithm.

The algorithm is based on a Gaussian search process (called Gaussian Adaptation) which is based on ideas from the natural evolution of species (Kjellström & Taxén, 1981; Kjellström & Taxén, 1992). Samples of the independent variables are randomly generated from a Gaussian \(n\)-dimensional distribution and evaluated for ‘success’ or ‘failure’. The outcome is regarded as a success if the sample belongs to a region \(R_A\) in \(R^n\) which is determined by an \(n\)-dimensional criterion function of the independent variables. Depending on the outcome of successive samples, the first and second order moments of the distribution is gradually aligned to \(R_A\).

The interesting point here is that the optimal convergence rate of the search algorithm is achieved if the probability of a sample to hit \(R_A\) is in the vicinity of 0.3 (actually \(1/e\)). At both extremes the convergence is zero. A hitting probability of 0 means that each sample falls outside \(R_A\), while a hitting probability of 1 means that every sample falls inside \(R_A\). In both cases, no information of how to modify the Gaussian distribution is achieved. Thus, in this application the order-disorder balance is crucial. In Kjellström & Taxén (1981, p. 707) the convergence rate as a function of the hitting probability is shown. This function has the same structure as the function in Figure 20.

In summary, rules and stabilizing structures are necessary in any social situation. The existence of a stabilizing core in a context means that each praxis develops its own stabilizing core. However, different forms of praxis in general have to interact or cooperate. Since

\[\text{Figure 20. Adaptability of an organization as a function of enterprise control}\]

\[^{19}\text{The issue of translation between stabilizing cores becomes obvious in strategies for outsourcing or in-sourcing parts of the business in an organization.}\]
every praxis develops its own core, this implies that interaction presumes some kind of translation between the core structures in each praxis. This is one reason why it is necessary to perform a translation and interpretation act when passing between different forms of praxis as described in Section 3.3.3.

It is important to note that the rules and structures as such do not impede coordinated action. Rather, they are prerequisites for such actions, provided they yield a proper balance between order and disorder. For coordination this becomes important when for example balancing between what should be coordinated from a corporate point of view and what can be left to local organizations within the organization to coordinate.

3.3.8 Tool usage

It goes without saying that the usage of tools is an intrinsic part of human activity:

“When the little child performs the perfect grip of the tweezers she starts her path as a tool maker; she becomes Homo faber, man as craftsman, the tool making animal” (Liedman, 2001, p. 180)

When the hominids started to walk upright, the hands were set free for other purposes than support for moving around. This was a decisive step in the evolution from ape to man. It opened up for an intricate interplay between the human organism and her environment which is still going on. Through hundreds of thousands of years of labour, the human hand evolved into a tool which is infinitely more sophisticated than that of any simian. The first tools were created and controlled by the hand, but the opposite is also valid: “... the hand is not only the organ of labour; it is also the product of labour” (Engels, 1954/1876). Physical and mental capabilities evolved together with the usage of tools, language and society.

In this process, objects in the environment were transformed into tools. According to Liedman (2001, p. 171) four different dimensions of knowledge can be found:

- The ability to produce tools.
- The ability to understand other people and the society in which they live.
- The ability to speak and understand utterances as well as being guided by a world of symbols.
- The ability to write and interpret writing.

Ever since Homo sapiens appeared, all her knowledge has been related to these dimensions in various blendings. None of the dimensions are however completely absent. When natural science was developed it did so hand in hand with tools, instruments and technology in general. The history of natural science is intertwined with the history of technology. Technology created a world which became the object of study for science, and scientific principles penetrate the use and development of technology.

In summary, the importance of the ability to make and use tools for the purpose of supporting coordination is evident.
3.4 The Activity Domain Theory - structured praxis

Our analysis of the praxis perspective started with a discussion of some salient features of praxis and in particular the dialectical relation between the individual and society. Next we defined a stratified model of the dialectical relation which is anchored both in the individual and in society. In this model, signs are mediating the dialectical relation in all strata. We proposed a set of coordination constituents which we conjecture is fundamental for coordinating socio-human activity.

In this section we will utilize the coordination constituents to define a structured form of praxis which we call activity domains (AD). The main purpose of introducing ADs is to find a representation of the coordination of socio-human activity which is suitable for analytical and constructive purposes. Thus, ADs should exhibit those features which we believe are fundamental for coordinating socio-human activity. This means that we are looking for a fundamental structure which is valid for coordinating human activity in general. For example, both a car repair shop and the development of complex systems should be possible to analyze by the AD construct.20

From the coordination constituents we define an activity domain by the following features:

- **Socio-human activity:** An AD is a structured form of praxis where actors are working together. The actors have a motive for coming together, which is the reason why the AD exists. The actors work on an object to produce a certain outcome21. The object is the main driver for organising the AD.
- **Signification:** The actions in the AD are mediated by signs which signifies what phenomena are perceived as relevant in the AD. The ‘reality’ perceived in an AD is always dependent on human interpretation and construction. Understanding is possible only through signs.
- **Intersubjectivity:** In order to achieve the outcome, intersubjectivity is necessary. Intersubjectivity arises in the interaction between humans and is a prerequisite for the emergence of various sign systems, above all language. Intersubjectivity is a prerequisite for shared meaning.
- **Contextuality:** In an AD actions are focused and situated.
- **Domain Transition:** Different ADs interact with each other. When doing so, a translation between the ideologies as manifested in the stabilizing cores may be necessary.
- **Experiential learning:** The capabilities and knowledge of actors in the AD are achieved in the iteration between reflection and action.
- **Orientation:** There is an orientational structure in the AD. This structure signifies what phenomena actors perceive in the AD and how these phenomena relate to each other. This means that the orientational view of the AD defines the context of the same domain (phenomena, relations between them, focal situations and situational horizon, see Section 3.3.3 Contextuality).

---

20. This means that we will regard the AD as the basic Unit of Analysis in the study (see Section 2.1.3 A single-case, multiple-units of analysis study).

21. These concepts are taken from Activity Theory, see for example Engeström, 1999. A short overview of the Activity Theory is given in Section 3.5.3.
THE ACTIVITY DOMAIN THEORY

- **Temporal structure**: There is a temporal structure in the AD. This structure signifies how actions in the AD are coordinated. The orientational and temporal structures interact with and influence each other.
- **Stabilizing core**: The actors in an AD share a common ideology, by which we understand any wide-ranging systems of beliefs or ways of thought. This ideology stabilizes the activities in the AD and provides a mechanism to balance between order and disorder in the AD.
- **Tool usage**: In an AD tools are used to support the activities.
- **Motive, outcome and object**: There is a motive in the AD which is the reason why the AD exists. In the AD actors work on object(s) in order to produce an outcome.

We will refer to the ensemble of constituents making up a particular AD as the *constitution* of that AD. The structural properties of an AD can be illustrated as in Figure 21:

![Figure 21. The structure of activity domains](image)

It must be emphasized that our conception of ADs highlights only what we mean are basic coordination aspects of socio-human activity. Other aspects would have been possible to include, such as the stability of the AD, conflicts between actors, turnover of actors, changes in the environment of the AD affecting the object and outcome, the maturity of the knowledge among the actors, etc.

### 3.5 Alternative perspectives on human activity

We have proposed the praxis perspective as a viable approach to the coordination problem, where coordination is regarded as a form of socio-human activity. Obviously, there are several alternative perspectives which could have been used as a point of departure. A relevant question is therefore why we have not departed from any one of these.

To this end we will shortly characterize and analyze some existing perspectives which we understand are similar to our basic ontological and epistemological assumptions. These perspectives are Structuration Theory, Actor Network Theory, Activity Theory and Organisational Semiotics. This selection of approaches must necessarily be subjective. Other possible
perspectives are ethnographical approaches, Pragmatism (Wittgenstein, Dewey), The Language Action Perspective approach, the Soft Systems Methodology just to mention some.

Our discussion of the praxis perspective indicates that we shall look for the following characteristics:

- The mutual interaction between the individual and society,
- interaction as the genesis of knowledge,
- intersubjectivity,
- individual cognition,
- semiotics and signs,
- experiential learning,
- contextuality,
- orientation,
- temporality,
- stabilizing core,
- and tools and artifacts.

We will include ‘domain transition’ in ‘contextuality’ for short since they are closely related phenomena. We will also try to evaluate their explanatory and constructive potential to inform the coordination of complex system development tasks. It is evident that our treatment of the alternative approaches can be only superficial and not inclusive. However, we believe that this sketchy walkthrough has its merits as a general background for positioning the praxis perspective.

3.5.1 Structuration theory

Structuration Theory (ST) was developed by Giddens (see e.g. Giddens, 1984)\textsuperscript{22}. ST is a comprehensive theory which seeks to understand how human agency, in the sense ‘capacity for accomplishing changes’, is related to social structures. In ST, the focus of social theory is social practices, not individual action, nor the existence and constraints of some kind of societal totality. It is social practices which constitute both individuals and society.

In traditional social sciences there has been a sharp division between those who regard social phenomena to be products of the actions of human ‘agents’ (e.g. Weber, 1983), and others who regard them as caused by the influence of social structures (e.g. Durkheim, 1972). Instead of regarding them in a dualistic way as separate and opposite things, Giddens proposes that structure and agency should be viewed as a mutually interacting \textit{duality}. The notion of ‘duality of structure’ in ST refers to the fact that structure is not simply constraining, but is also enabling human action. The actions of human agents discursively and recursively form the sets of rules, practices and routines which, over time and space constitute structure. This process is called ‘structuration’.

\textsuperscript{22} The survey of ST is mainly collected from Jones (1999) and Rose (1998).
When structure is being continuously produced and reproduced through action, this leads to another significant aspect of structuration, that of routinization. Routine is fundamental to the continuity of the personality of agents and to the institutions of society. Individuals acquire ontological security through their daily engagement in familiar routines and encounters. These encounters are also constitutive of social institutions which means that they enable the continuity of social life and the creation of order over time and space. From this follows also that time and space are important elements in ST.

One of the most controversial tenets of ST is that the social structures do not exist independent of human action. Giddens refers to the structures as ‘memory traces’ in the human brain which means that they are not material entities. Thus, there are no material elements in ST, only elements in the heads of the actors.

The insight that information systems are closely associated with social and human issues has inspired a number of researchers to use ST as a starting point for theoretical development, analysis and operational purposes (see e.g. Jones, 1999 or Rose, 1998). Since ST is a very comprehensive theory it has lead to a number of different appropriations and interpretations of it. Mostly ST has been used for analytical purposes, i.e., as an analytical framework for the understanding of empirical situations or cases. The usage of ST for operational or constructive intents has been limited. One obvious difficulty is that ISs have a technological, material character which is hard to conciliate with the non-material stance of ST. The attempts to adapt structuration to incorporate the material aspects of IS have encountered a number of serious problems which have not been resolved.

**Evaluation**

Obviously, ST has much in common with the praxis approach taken in this thesis. Contextuality, temporality, orientation and stabilizing core are all important elements of ST as well as experiential learning in the form of recursive constitution of routines and structure. The notion of ‘dualism’ recognizes that a dialectical attitude is important. The mutual constitution of agency and structure is similar to the notion of the socio-human reality in praxis.

However, in praxis there is an objective, material aspect which is not present in ST. In our elaboration of praxis the material aspect is related to the cognitive system of individuals through the sign. Anything material may take on a semiotic value where the signifier and the signified are associated in the brain. We claim that the ‘memory traces’ in ST are in fact equal to this association. Thus, ST recognizes the cognitive side of the sign whilst ignoring the physical side of it. Obviously, this is the weak point in ST when it comes to constructive purposes of artifacts like IS. This is also confirmed by the research community so far. The praxis approach on the other hand has resulted in practical, concrete implementations of IS.

### 3.5.2 Actor Network Theory

Actor Network Theory (ANT) is one strand of research dealing with the social construction of technology where social and technical phenomena are considered together. ANT is con-
cerned with the creation and maintenance of coexisting networks of human and non-human elements. Instead of apprehending the social and technical phenomena as separate elements, ANT treats them as inseparable. Latour (1998) argues that people and artifacts should be analyzed with the same apparatus. There is a need for new methodological and theoretical devices that we can use to analyze hybrids of people and artifacts such as for example information systems.

ANT examines the motives and actions of groups of actors who form networks of aligned interests. These networks include both humans and non-humans as actors. This standpoint has caused considerable controversy since it can be claimed that human responsibility vanishes. Non-human resources can ‘stand in or speak for’ or be delegates for particular viewpoints or truth-statements which help to maintain a particular network of alliances.

The empirical focus of ANT is to trace and explain the processes of creating and maintaining stable networks of aligned interests, or to examine why such networks fail to establish themselves. Thus, ANT is both a theory and method combined. It provides theoretical concepts as ways of viewing the world and suggests that these elements should be traced in the empirical work. The researcher is led to investigate and document network elements, both human and non-human, processes of translation and inscription, the creation of black boxes or immutable mobiles and the degree of stability and irreversibility of networks and elements.

Walsham (1997) has reviewed the status of ANT in IS research. Many researchers use concepts from ANT to analyze their findings, and the main point seems to be that ANT does not privilege either social aspects or technology. Some critique has been issued against ANT, for example that ANT addresses the local and contingent but pays little attention to broader social structures. However, Walsham’s overall conclusion is that ANT has much to offer in both theoretical and methodological terms. A combination of e.g. ANT and the Structuration Theory of Giddens might be fruitful.

**Evaluation**

As with Structuration Theory, ANT has a lot in common with the praxis approach, among all, the dialectical view of the interaction between human agents and artifacts. There is also a strong notion of context, orientation and temporality in ANT. Moreover, the constructs in ANT are obviously very suitable for analysis of the complicated interplay between networks of actors (human or non-human) over time and space. In short, ANT seems to be an interesting and useful methodology based on an extreme ontology. However, the view on non-humans as actors having motives, goals, intentions, etc., is alien to the marxist / praxis tradition, and this stance does not seem to be necessary for the theory as such.

Results concerning the usage of ANT for constructive purposes seem to be limited so far. A combination of the constructive power of our praxis approach with the analytical power of ANT might be an interesting perspective to elaborate.
3.5.3 Activity theory

Activity Theory (AT) has long historical roots that can be traced back to Kant, Hegel, Marx and Engels\(^2\). It was developed in the cultural-historical psychology school where the principal contributors were Vygotsky, Leont’ev and Luria. It also has relations to the Dewey pragmatism and the symbolic interactionism by Mead. During the 80’s and 90’s, AT has been further evolved by Finnish researchers, most prominently perhaps by Engeström.

AT does not accept a dualistic conception of an isolated, independent mind. Cognitive processes are not independent and unchanging ‘abilities’ or ‘functions of human consciousness’; they are processes occurring in concrete, practical activity. In AT the activity is the basic unit of analysis. A minimal meaningful context must be included in the analysis; this is the activity.

The object of research using AT is always collective even when we study individual actions. An activity is a form of doing directed to an object. Activities are distinguished from each other by their objects. Transforming the object into an outcome motivates the activity. The object may be material or intangible but it must be possible to share for manipulation by the participants in the activity. Real life activities can be distinguished by their object.

Relations between elements of an activity are not direct but mediated. The relationship between actors\(^2\) (the subject) and the object of an activity is mediated by a tool into which the historical development of the relation is condensed. The tool both enables and limits the actions of the subject in transforming the object. Historical experience is crystallized into it, but the subject is limited by the capabilities of the tool in manipulating the object. Signs and symbols are also included in this mediating relationship. Every activity has both an internal (properties of objects penetrate into the subject and transform her) and external side (subjects are transforming the object).

Engeström has revised the original AT theory by adding elements to it in order to incorporate social aspects. Community is added to form a triangle with three mutual relationships between subject, community and object. The relationship between the subject and the community is mediated by rules. The relationship between the object and the community is mediated by a division of labour (see Figure 22).

AT has so far been mostly used as an analytical tool. It has not been operationalised in the meaning that there is a gap between theory and concrete methods and tools to be used in practice.

---

24. The survey of AT is mainly collected from Engeström (1999a) and Kuutti (1996)
25. Note that the subject can be one individual or a group of individuals.
Evaluation

AT and the praxis approach obviously have common ontological and epistemological roots. As with praxis, AT tries to capture the totality of human activity. The elements in an activity (object, outcome, subject, community, tools, signs, symbols, division of labour, rules, etc.) are included in our appropriation of praxis. Experiential learning or continuous change are also common to both approaches.

However, there are also several differences between AT and the praxis approach. AT downplays the role of the individual since the subject can be both individuals and groups. Furthermore, in AT signs are not treated explicitly. Another difference is that the issue of context translation is not treated in AT. Finally, the use of AT for constructive purposes seems to be limited.

3.5.4 Organisational semiotics

Organisational Semiotics (OS) is defined by Stamper as informatics from a social angle (Stamper, 2001). This discipline deals with both technological, human and social aspects of information and will according to Stamper play a major role in the future. The reason why Stamper somewhat provocatively characterizes this discipline as “informatics without the computer” is that he wants to base his approach in a precise philosophical understanding of the information concept. Traditional computer informatics went wrong because of a failure to build upon primitive concepts whose meaning must be defined ostensively, i.e., by explaining the meaning of a word by ostension, that is, by pointing to something to which the word applies.

Stamper claims that semiotics leads to a more precise definition of information as various properties of signs. This will cross the bridge between language and reality. Semiotics can solve this problem since it builds on an ostensively definable concept: the notion of a sign. Stamper wants to advance semiotics to study sign products: shared knowledge, mutual commitments and institutions. The key problem is linking linguistic signs to the reality we are talking about. This is a problem both of ontology and epistemology.

Stamper means that we live in two more or less orthogonal worlds, the material and non-material worlds. Information creates the understanding and relationships from which organizations and communities are built. Information is the ‘material of our social world’. This leads Stamper to define a new ontology: actualism. The actualist ontology entails a profound shift in the way we talk about the world. The only reality we can know directly consists of here and now. Signs exist here and now. The past and future do not exist other by signs that represent them. The only knowable reality consists of affordances – invariant repertoires of behaviour, either substantial affordances or social norms. Some agent bears the responsibility for every feature of the world as we know it. Society is always the root agent for human knowledge and behaviour. This means that society is regarded a responsible agent. Other agents can be individuals or groups.

---

26. However, in a recent work Leiman discusses signs in relation to AT (Leiman, 1999).
THE ACTIVITY DOMAIN THEORY

In order to capture socially established meanings, Stamper defines a formalism to model invariant repertoires of behaviour. He calls this formalism Norma, a language of norms and affordances. A particular set of repertoires valid for an organization can be illustrated in a schema, where the nodes do not represent concepts in anyone’s head but socially shared, physical or social affordances. The Norma approach has been applied in a large number of organizations with very large cost savings.

Evaluation

The importance placed on the role of signs is a feature that Organisational Semiotics share with our praxis approach. From this follows that intersubjectivity is regarded in both approaches as a major characteristic of human activity. Intersubjectivity is the basis for the Norma language, and the fact that this language has been used with good results in practice demonstrates the importance of considering intersubjectivity in constructive situations. In ordinary analysis common sense often passes unnoticed which might create huge maintenance costs of IS when redesign becomes necessary.

Other similarities with the praxis approach are the importance placed on contextuality and tools (in the form of Norma).

A major difference between OS and the praxis approach is the importance placed on the individual. In OS perceptions and conceptions cannot be used since they are private to the individual in whose mind they occur. Thus “informatics has no need for perceptions and conceptions” (Stamper, 2001, p. 123). This means that OS can in be regarded as the extreme opposite to Structuration Theory which acknowledge only the existence of ‘memory traces’. Our praxis approach on the other hand considers both individual cognition and objective, shared meanings mediated by the sign.

Another difference is that human agency is downplayed. ‘Society’ can never be an actor in the praxis approach, not even in the Actor Network Theory. Experiential learning, temporality, orientation and context translations do not seem to be salient in the OS approach.
CHAPTER 3

3.5.5 Summary

In the table below, we try to summarize the different approaches. By ‘inherent’ we mean that the particular characteristic is a basic assumption in the approach. A ‘-’ indicates that this characteristic is not salient in the approach. Stars ‘*, **, ***’ indicate the salience of the characteristic, ‘****’ indicating the strongest presence.

Table 14. Comparison between different approaches

<table>
<thead>
<tr>
<th>Approach / characteristic</th>
<th>Structuration Theory (ST)</th>
<th>Actor Network Theory (ANT)</th>
<th>Activity Theory (AT)</th>
<th>Organisational Semiotics (OS)</th>
<th>Praxis Approach (PA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mutual interaction individual - society</td>
<td>inherent</td>
<td>inherent</td>
<td>inherent</td>
<td>inherent</td>
<td>inherent</td>
</tr>
<tr>
<td>Interaction as genesis of knowledge</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>inherent</td>
</tr>
<tr>
<td>Intersubjectivity</td>
<td>-</td>
<td>inherent</td>
<td>inherent</td>
<td>inherent</td>
<td>inherent</td>
</tr>
<tr>
<td>Individual cognition</td>
<td>inherent</td>
<td>inherent</td>
<td>inherent</td>
<td>inherent</td>
<td>inherent</td>
</tr>
<tr>
<td>Semiotics, signs</td>
<td>inherent</td>
<td>inherent</td>
<td>*</td>
<td>inherent</td>
<td>inherent</td>
</tr>
<tr>
<td>Experiential learning</td>
<td>inherent</td>
<td>inherent</td>
<td>inherent</td>
<td>inherent</td>
<td>inherent</td>
</tr>
<tr>
<td>Contextuality</td>
<td>***</td>
<td>***</td>
<td>***</td>
<td>***</td>
<td>inherent</td>
</tr>
<tr>
<td>Orientation and temporality</td>
<td>inherent</td>
<td>inherent</td>
<td>**</td>
<td>inherent</td>
<td>inherent</td>
</tr>
<tr>
<td>Stabilizing core</td>
<td>inherent</td>
<td>inherent</td>
<td>inherent</td>
<td>inherent</td>
<td>inherent</td>
</tr>
<tr>
<td>Tools and artifacts</td>
<td>inherent</td>
<td>inherent</td>
<td>inherent</td>
<td>***</td>
<td>inherent</td>
</tr>
<tr>
<td>Explanatory purpose</td>
<td>*</td>
<td>***</td>
<td>**</td>
<td>**</td>
<td>-</td>
</tr>
<tr>
<td>Constructive purpose</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>**</td>
<td>***</td>
</tr>
</tbody>
</table>

All approaches share the ontological position of the mutual interaction between individual and society as the formation of human activity. However, none of the approaches except the PA have a clear focus on the interaction as the genesis of knowledge. Intersubjectivity is salient in all approaches except ST. Only in ST and PA individual cognition is salient. OS and PA are the only approaches that focus specifically on signs. Experiential learning and change seem to be inherent in all approaches except OS. Contextuality is also strong in all approaches although only PA highlights the context translation issue. Orientation and temporality are also salient in all approaches except OS. Notions of a stabilizing core are salient in all approaches. Tools and artifacts are important in all approaches except ST. The ANT is strongest on the usage for explanatory purposes. So far the PA approach has not been demonstrated for this purpose. Finally, the PA and OS approaches seem to be the strongest ones for constructive purposes.

This analysis is grounded in a superficial reading of the literature and thus the results should be treated accordingly. However, we claim that the analysis shows that not all of salient characteristics of the praxis approach are present in another approach taken one by one. If we conjecture that all these characteristics are important for explanatory and constructive purposes in connection with the coordination of the development of complex systems, then the praxis approach may be a good starting point. The ultimate test of this conjecture will be its impact in practical applications.
3.6 Summing up

In this chapter we have outlined a theory for socio-human activity which originates in the praxis perspective. The theory tries to encompass both individual / subjective aspects as well as social / objective aspects of human activity. In particular, the theory emphasizes the dialectical relation between the individual and society as the genesis of human capabilities and knowledge. The reality as perceived by actors is mediated by signs which have a physical signifier aspect and a signifying aspect. The signifier and the signified are associated in the brain of humans. Thus our theory presumes that there exists an objective world which is independent of, but related to human beings. However, human activity changes that world which in turn changes the perceived reality of humans.

Next we proposed a set of coordination constituents which we argue are fundamental for coordinating socio-human activity. We tried to ground these constituents in the cognitive and linguistic literature, thus underpinning their fundamental character. These constituents are utilized in the structuring of praxis into activity domains, which we consider to be a fundamental form in which the coordination of socio-human activity can be moulded. The intention of defining activity domains is that they should be a suitable basis for constructive and explanatory purposes in connection with the coordination of the development of complex systems. This will be treated in the following chapters of the study.
CHAPTER 3
4 Articulating Coordination

In this chapter, we will define our understanding of coordination. Although coordination is easily recognized when it breaks down, there appears to be no commonly accepted way of defining coordination. For example, Larsson (1990) lists nineteen different definitions and Malone & Crowston (1994) report on eleven definitions. Malone & Crowston also emphasize that the study of coordination must draw on a variety of disciplines including organization theory, management science, computer science, economics, linguistics and psychology (ibid., p. 88). Melin (2002) adds the area of informatics as another relevant discipline.

Most definitions of coordination include human and social aspects as well as technological ones, for example actors, actions, goals, purpose, meaning, communication, cooperation, collaboration, processes, information processing, information systems, etc. Furthermore, interaction and interdependency are emphasized. One example is Larsson who defines coordination as:

[The] adjustment of interdependent parts to one another influencing their interactive outcome (Larsson, 1990, p. 72)

Larsson (ibid., p. 6 ff.) also describes some deficiencies in the understanding of coordination:

- The emphasis on formal coordination mechanisms together with the separation from the social ordering aspect of coordination have resulted in a lack of consideration of informal coordination. This emerges from repetitive interactions between members creating uniform states of mind which crystallize into mores, norms, institutions, etc.
- The design orientation of the coordination mechanisms focuses on primarily pre-planned coordination. This leads to a large separation in time between the design of coordination mechanisms and the application of these mechanisms. This may be inadequate in situations characterized by uncertainty, instability and complexity.
- The focus on top-down coordination leaves to a large extent out self-coordination by the actors. Self-coordination is useful since it can create well-adjusted on-the-spot coordination in unforeseen situations where pre-planned coordination fails.
- The traditional conceptualization of coordination focus on concrete, material interdependencies in terms of work input-output flows. However, there are also abstract inter-dependencies that need to be coordinated, like information flows, perceptions, and meaning relations.

Malone & Crowston (1994) discuss a research agenda for a coordination discipline. Some issues that should be addressed are:

- how to identify and analyze specific coordination processes and structures,
- how to represent coordination processes,
- how to classify different coordination processes,
- how generic coordination processes are,
- what kind of dependencies there are.
Malone & Crowston also state that there are practical needs which should be solved such as designing tools that enable people to work together more efficiently and finding more efficient, flexible and satisfying ways to organize human activity (ibid., p. 111).

From the discussion in the literature we can identify the following needs concerning coordination:

- Interaction and interdependencies must be considered.
- Coordination cannot be confined to a rational planning and controlling process only. The social interplay between actors leading to conceptualization, shared meanings, norms and mores must be considered as well.
- The gap between the design and use of coordination mechanisms must be narrowed.
- Coordination should be distributed and not confined to a top-down approach.
- Coordination processes and structures must be identified.
- The coordination mechanisms shall be operational; that is, they should provide methods and tools to meet practical needs.

We want our conception of coordination to be compliant with these needs. This means, that in the evaluation of the impact on coordination from the Framework intervention, we should consider these needs. This is done in Section 9.2.1 Impacts on coordination.

### 4.1 Coordination apprehended as an activity domain

The term ‘coordination’ may stand for different things. In daily ‘language games’ we do not have to reflect on this as long as the usage of the term results in the intended consequences. However, in scientific contexts we need to be more precise. Goldkuhl (2002) emphasizes the importance of a linguistic determination of the terms used, for example:

- To coordinate different activities (a verb)
- Coordination as a result of activities (a noun).
- A coordinated system (an attribute or state).

All these determinations of the term ‘coordination’ are present in our conception. It is our intention that the specific use of the term will be evident from the text.

We will start from the definition of coordination according to Malone & Crowston:

> “Coordination is managing dependencies between activities” (Malone & Crowston, 1994, p. 90)

This definition indicates that something is done by someone. Coordination is an activity in itself which provides coordination of other activities. Thus coordination can be regarded as a meta-activity, an aspect which is emphasized in Melin (2002). Coordination activities may be called ‘management’ activities in order to separate them from ‘productive’ activities (Melin, 2002, p. 62, referring to Iden, 1994). We prefer calling these ‘non-transformational’ and ‘transformational’ activities respectively. These activities should be regarded as ‘ideal types’ (Weber, 1983) in the sense that they represent two distinct types of activities. In practical settings the distinction between activities is not always clear-cut.
ARTICULATING COORDINATION

Transformational activities transform objects from one condition to another, while non-transformational activities are evoked to manage the transformational activities. Thus, coordination may be seen as a non-transformational activity which manages the dependencies between transformational activities.

In complex development situations such as the one at Ericsson, several types of transformational activities are needed such as software design, hardware design, radio design, etc. These various activities may be regarded as carried out in different activity domains which need to interact in order to develop the complete product.

From this the pivotal step in our elaboration of the coordination concept follows: we define an activity domain associated with the development activity domain: the coordination domain. The coordination constituents in these two domains are the same but utilized for different purposes. In the development domain they are used to coordinate the development. The object for this activity is the telecommunication system. In the coordination domain the purpose is to provide coordination to the development domain. In this domain the elements become the objects of the activity. This means that the coordination domain develops the coordination constituents which are used for coordination in the product development domain. In order to distinguish these domains we refer to them as the ‘coordination domain’ and ‘usage domain’ respectively. The crucial insight in this arrangement is to realize that the coordinating constituents are the same. However, they appear in two different contexts in which different properties of the constituents may be brought out.

An example might clarify: consider a piano concert where a performer plays the piano in front of an audience. The outcome of this activity is an immediate musical experience in which the artist uses the score to coordinate her performance. The score may be seen as an expression for temporality since it prescribes the order in which the keys are pressed. This activity corresponds to the usage domain. However, for the composer the score is the outcome of his activity. Thus, this activity corresponds to the coordination domain in our terminology. The object, the score, is the same in both cases but in the concert activity it is used while it is developed in the composing activity.

An interesting consequence of this way of apprehending coordination is that the evolution of the coordination constituents may be done in close interaction between developers and users. In the example above this would correspond to a close cooperation between an artist and a composer in composing a new score. In the Domain Construction Strategy (Section 5.1.6, page 114) in the Framework a continuous pendulation is taking place between the coordination domain and the usage domain. This means, for example, that an evolutionary IS development approach is used in the Framework.

1. More precisely, it is the derived elements in the Framework, such as conceptual models, process models, information system support, etc. which are developed (see Chapter 5 The structure of the Framework).
Based on these considerations we propose the following conception of coordination:

Coordination is an activity domain of its own which we call the coordination domain. The motive of the coordination domain is to provide coordination of the transformational activities in an associated activity domain, which we will call the usage domain. The object of the coordination domain is the coordination constituents. Thus, the coordination constituents are designed in the coordination domain and used in the usage domain.

This conception is illustrated in Figure 23:

![Figure 23. The interaction between the coordination and usage domains.](image)

This means that the only difference between the coordination domain and the usage domain is the change in focus between designing and using the models in the Framework and their implementation in the IS. Some actors may participate in both domains, for example a project manager\(^2\).

One consequence of defining coordination in this way is that every activity domain can be conceived as having an associated coordination domain. This means for example that what is apprehended as transformational activities in one domain may be the result of non-transformational activities in other activity domains. For example, the coordination of increments or work packages (see Section 7.2.1 Coordination situations) in the project management activity domain may be the result of coordinating software build files in a software development activity domain.

Thus we apprehend coordination as a socially organized human activity where only aspects relevant for the coordination purpose are considered\(^3\). All other aspects are subdued. This means that we can reduce the number of salient dimensions in the coordination domain to a minimum for coordination purposes such as name, revision, relation, some attributes, etc.

---

2. This was the case in both the S and A-domains. However, in the C-domain this gap between design and usage had widened and no project manager was present in both domains. This increased the separation in time between the design of coordination and the usage of it. Thus, the deficiency identified by Larsson (1990) was boosted rather than decreased.

3. We will limit our discussion to coordination related to human activity. In Malone & Crowston (1994), a more general approach to coordination is taken which includes computational systems, biological systems, etc.
ARTICULATING COORDINATION

The coordination domain itself may be divided into several coordination situations, each with a particular coordination focus. These coordination situations are in fact structured as activity domains of their own, but in order not to confuse things we will call them coordination situations (see Chapter 7.2.1 Coordination situations).

4.1.1 Motive, outcome and actors

The motive why the coordination domain exists is that transformational activities must be coordinated. The actors are individuals in roles which impact or are impacted by the coordination. In the Ericsson practice, these may be product managers, project managers, configuration managers, test managers, designers, etc. Some of these actors may work in both the coordination and usage domains, for example project managers. The outcome of the coordination domain is the coordination of the associated usage domain, for example that of complex systems’ development.

The nature of the outcome depends on what the actors in the usage domain perceive as important coordination aspects. Some examples of such aspects from the Ericsson practice are:

- Shared meaning concerning the meaning of coordination,
- transparency in perceiving the coordination context,
- traceability between items to be coordinated,
- dependencies between items to be coordinated4,
- project planning, monitoring of progress,
- global access to coordination information,
- controlling changes and error correction,
- time to perform repetitive coordination tasks,
- number of information systems needed for coordination,
- number of interfaces needed between information systems, etc.

Thus the outcome is constructed in the development practice by actors which may appear in both the usage and coordination domain. This means that the coordination domain is socially constructed by the actors in that domain. One consequence of this is that coordination domains serving different usage domains will be constructed differently. This can be seen clearly from the experiences of using the Framework at Ericsson. For example, the Conceptual Models in the S-domain (see Figure 41, p. 139) and the A-domain (see Figure 42, p. 145) are quite different.

The construct of reinforcement rods as defined in Chapter 8 is an attempt to elicit elements, for example ‘seeing dependencies’, which are important for successful outcomes of development projects. This means that the construction of the coordination domain should be done in such a way that the reinforcement rods related to coordination are supported.

4. Malone & Crowston point out that identifying various kinds of dependencies and the processes that are used to manage these dependencies are important to understand coordination (Malone & Crowston, 1994, p. 91).
4.1.2 Coordination items

The object of the coordination domain is the elements of the Framework which are derived from the coordination constituents in the Activity Domain Theory. These elements are the Conceptual Model, the Process Model, the Transition Model, the Stabilizing Core and the Information System (see Chapter 5).

In the coordination domain, only items that are somehow impacted by the coordination activity are significant. Moreover, these items are characterized only from a coordination point of view. For example, a product may be transformed from a state ‘preliminary’ to a state ‘verified’ by some transformational activity. From a coordination point of view it is not interesting how this transformation is carried out, only that the product has changed its state from ‘preliminary’ to ‘verified’. The product is characterized by an identity, state and possibly some attributes. Other characteristics of the product are not relevant in the coordination domain.

When objects appear in the coordination domain and are characterized accordingly, we will refer to them as coordination items. The properties of these items and their relations to other items are defined in the Conceptual Model in the Framework (Chapter 5.1.1), while the state transitions are defined in the Process Model.

Examples of items involved in coordination in the Ericsson practice are:

- The product and its describing documents.
- Requirements on the product stated by the customer.
- The specification of the product in terms of functionality and other characteristics.
- Incremental development steps during the project where parts of the complete specification are fulfilled.
- Test cases which are used to test that the product fulfills its specification.
- Trouble reports describing various types of errors.
- Engineering change orders calling for controlled changes of the product.
- Baselines which describe a certain achieved configuration of the product which cannot be changed without specific control procedures.
- Milestones describing the progress in the project.

4.1.3 Shared meaning

The actors in a coordination domain need to achieve some shared meaning concerning this domain. They need to agree upon what coordination items are important, how these are to be characterized and how they are related to each other. They also need to agree about the dependencies among the transformational activities and rules for how manage them. In fact, they need to agree on all the aspects of activity domains. Thus, what holds true about the coordination domain is by and large socially agreed. This is also emphasized by Malone & Crowston:

5. Called Change Requests at Ericsson.
ARTICULATING COORDINATION

“In order to analyze a situation in terms of coordination, it is sometimes important to explicitly identify the components of coordination in that situation. [...] It is important to realize that there is no single ‘right’ way to identify these components of coordination in a situation. (Malone & Crowston, 1994, p. 101)

There is a close connection between the shared meaning, communication and language: “How [...] can actors establish common languages that allow them to communicate in the first place” (Malone & Crowston, 1994, p. 99). This is also pointed out by Goldkuhl & Röstlinger:

“A practice is coordinated through communication. Different linguistic actions are necessary in order to coordinate actions so that the intended result can be produced. This is necessary within a practice in which several producers cooperate” (Goldkuhl & Röstlinger, 1998, p. 5, my translation)

The importance of communication for establishing shared meanings in coordination situations is also stressed by Melin:

“Communication about other actions (of coordinating and coordinated character as well as executed or planned) is important to the establishment of a common understanding in the actual coordination situation.” (Melin, 2002, p. 402, my translation)

The capacity of an organization to manage complex patterns of interdependent activities is closely related to its capacity to manage the communication required for coordination (March & Simon, 1958, p. 183). The efficiency of the communication is increased by using few symbolic and verbal conventions in order to provide large amounts of information. An example is a blueprint whose symbols have a shared meaning to the actors in the organization (ibid., p. 184).

Furthermore, March & Simon point out that the language needed for communication is well developed when it refers to concrete things or things that can be easily classified, that is, things that have acquired a state of ‘facts’ in the organization. When this is not the case the communication is much harder:

“... it is extremely difficult to communicate about intangible objects. Hence, the heaviest burdens are placed on the communications system by the less structured aspects of the organization’s tasks, particularly by activity directed towards the explanation of problems that are not yet well defined.” (March & Simon, 1958, p. 186)

This is but all to well illustrated by the attempt to achieve consensus about the meaning of the seemingly well known phenomena ‘requirement’ in the discussion related in Section 7.5.1 The agony of achieving shared meaning - an example, p. 192.

6. In this sense ‘activity domains’ and ‘practice’ are comparable
4.1.4 Orientation

The context of the coordination domain is defined by the coordination items which the actors consider are important to the coordination outcome. The orientational structure is the static characterization of that domain. In this structure the coordination items and the static relationships between them are found. Other important characteristics are attributes on coordination items and relations, state sets for coordination items. The orientational structure is often represented by various models like information models, business models, data models, etc.

4.1.5 Temporality

Temporality concerns the dependencies between the transformational activities as apprehended in the coordination domain and is thus compliant with the definition of Malone & Crowston (1994, p. 90). Examples of temporal structures are business processes, development processes, etc.

It is evident that the orientational and temporal structures are interdependent. What is not so clear is the nature of this interdependency. Malone & Crowston claim that it is not necessary to consider these dependencies in the definition of coordination:

“More generally, there are many types of dependencies between objects that are managed by coordination processes. For instance, an important part of managing the design of complex manufactured products involves managing the dependencies among different subcomponents. At first glance, our definition of coordination (as managing the dependencies among activities) might appear to omit dependencies among objects that are not activities. We believe, however, that this focus has the advantage of greatly simplifying the analysis of a coordinated situation. In fact, it appears that all dependencies that require coordination can be treated in this way.” (Malone & Crowston, 1994, p. 97).

However, for the purpose of coordinating the development of complex systems we claim that it is equally important to consider the dependencies between the objects. For example, in a system such as the one illustrated in Figure 2 Dependencies between development tasks in a network node, the functional dependencies put restrictions on the dependencies between activities. One obvious example is that the complete verification of an increment cannot be completed before other increments in the dependency chain are completed.

4.1.6 Domain transition

As described earlier, the usage domain needs to interact with other activity domains in order to achieve its outcome. Consider the example in Section 4.1.2, page 100. The two significant states of the product in the coordination domain are ‘preliminary’ and ‘verified’. In the transformational activity domain these may be called something else, for example ‘started’ and ‘approved’. The reason for these different terms for the same phenomenon may be different languages, traditions, rules, norms, etc., in the two domains. Thus an interpretation must be done in order to translate one term into the other.
Moreover, the state ‘verified’ of the product may be a function of a number of states on related items, for example documents describing the product. This implies that there must be some rule which assigns product states from document states.

Thus, in general, the interaction between two domains includes both an interpretation and a mapping between the elements. This interaction has both social aspects such as linguistic ones and more formal, technical ones such as for example rules or protocols specifying the interaction between two information systems. In order to reflect all these aspects we will refer to the relation between two domains as a domain transition which also draws on the connotation that actors can move between domains.

4.1.7 The stabilizing core

The stabilizing core of an activity domain refers to stabilizing elements such as standards, rules, routines, norms, etc. Without these stabilizing elements no coordination is possible. They have the function of “... reducing the infinite number of things in the world, potential or actual — to a moderate number of well-defined varieties” (March & Simon, 1958, p.181). Examples of such elements from the Ericsson practice are product and document identities which follow well-established rules.

There is a close connection between the scope of the stabilizing core and the capability of the activity domain to adapt to changes. If the stabilizing core is too encompassing, the activity domain will adapt slowly, if at all. In spite of changing circumstances the actors will continue to pursue their work according to rules which might be counterproductive for the domain.

On the other hand, if the stabilizing core is too small, the activities in the activity domain cannot be coordinated and the domain is unable to adapt to the new circumstances. The key insight here is that some stabilizing elements are necessary if the activity domain shall be capable of adaptation: “... in order for an organization to behave adaptively, it needs some stable regulations and procedures that it can employ in carrying out its adaptive practices.” (March & Simon, 1958, p.119).

From the viewpoint of the coordination domain, the activity in this domain concerning the stabilizing core is to establish the core in such a way that a proper balance can be upheld between order and chaos in the activity domain.

4.1.8 Tool usage

The tool aspect of the coordination domain concerns the establishment of a proper tool support for coordination, mainly various types of information systems and their interaction. These ultimate purpose of these tools is to support the management of the transformational activities: “One possibility would be computer-based tracking systems that made it easy for

7. In this sense ‘activity domains’ and ‘organizations’ are comparable.
everyone in the project to see status information about all the activities and their dependencies” (Malone & Crowston, 1994, p. 94).

4.2 Summary
In this chapter we have defined our understanding of the coordination concept. Coordination is apprehended as an activity domain, where only aspects impacting coordination are relevant. In the next chapter we will see how a coordination domain may be realized in practice.
5 The structure of the Framework

In this chapter we will describe the Framework and the elements it consists of. The purpose of applying the Framework is to construct coordination domains as discussed in Chapter 4 Articulating Coordination.

As described in Chapter 6 The history of the Framework, the structure of the Framework evolved during a period of more than 10 years in the Ericsson development practice. The Framework consists of three models, an information system (IS), a stabilizing core and a strategy for domain construction. Each element is derived from one or several coordination constituents (see Chapter 3.3.1 Coordination constituents – a conjecture). The Framework is designed in a canonical way. This means that the elements in the Framework should be as few as possible but still encompass all the coordination constituents. This explains why there are three models and not, for example, two or four in the Framework. In principle it is possible to conceive of other structures of the Framework. However, this has not been investigated in the study.

The Framework should be regarded as a coherent whole in the sense that all elements in the Framework are important to the construction of coordination domains. The term ‘framework’ refers to the fact that the elements in the Framework and their interactions are specified. However, the particular implementation of these elements in the Ericsson practice may be exchanged for other implementations fulfilling the same specifications. For example, the information system eMatrix used in the Framework may be replaced by another information system provided that the evolutionary way of working inherent in the Framework is retained.

The term ‘construction’ has two connotations. Firstly, the Framework is meant to be operational in system developing organizations. This means that the elements of the Framework have to be constructed in a particular setting by the actors in the coordination domain. Secondly, ‘construction’ refers to the social aspect of coordination. What is considered to be the reality concerning coordination emerges as a result of the social interaction among the actors in the coordination domain. This does not imply that coordination may take any form. On the contrary, the meaning of coordination is confined by concrete cultural, historical, organizational and technological circumstances. The result of applying the Framework will be a coordination domain which includes shared meaning among actors, instantiated models and instantiated IS support.

5.1 The Framework elements and how they are related

As defined in Chapter 4 Articulating Coordination there are two activity domains involved in the construction of coordination. One is the usage domain which produces, for example, telecommunication systems by coordinating a number of transformational activities. The other domain is the coordination domain which produces ‘coordination’ which is used in the usage domain. This means that the models and the information system appear in both the usage domain and the coordination domain but in different roles. In the coordination domain they are the working object and the outcome of the activities. In the usage domain they are
CHAPTER 5

means to coordinate the activities to produce another outcome such as a telecommunication system.

The Framework elements are defined as follows:

- The **Conceptual Model** signifies the structure of the coordination domain in terms of coordination items and their static relationships. One example of this is that “requirements are allocated to design items”. The Conceptual Model is derived from the contextuality and orientation constituents in the Activity Domain Theory. An example of a Conceptual Model is given in Figure 38, p. 131.

- The **Process Model** signifies the dependencies between activities impacting the coordination items, for example that “requirement prioritization” comes before “requirement allocation”. This model corresponds to the definition of coordination given by Malone & Crowston (1994, p. 90). The Process Model is derived from the temporality constituent in the Activity Domain Theory. An example of a Process Model is given in Figure 40, p. 132.

- The **Transition Model** signifies how the coordination activity domain interacts with other activity domains, that is how coordination items are interpreted and translated when they appear in other contexts. For example, out of all the information produced in a software design context only some is important in the coordination domain, e.g., when a source code file has changed its status or revision. The Transition Model is derived from the domain transition constituent in the Activity Domain Theory. An example of a Transition Model is given in Figure 2, p. 3.

- The models are implemented in the **Information System (IS)** and tried out in the usage domain. The Information System is derived from the tool usage constituent in the Activity Domain Theory. An example of a view from the system used in the Framework is given in Figure 39, p. 132, where instances of the types defined in the models can be seen.

- The **Stabilizing Core** consists of rules, norms, standards, etc., which provide the necessary stability in the domain. The Stabilizing Core is derived from the stabilizing core constituent in the Activity Domain Theory. One example from the Ericsson practice is the ‘Corporate Basic Standards’ which is a set of rules for how to identify and version control products and documents.

- In order to achieve a shared meaning about the coordination domain a strategy is included in the Framework, the **Domain Construction Strategy**. This strategy consists of an ongoing iteration between reflection and action phases in which the models and the IS are alternatively reflected upon and tried out in practice. In this process the IS is continuously changed. Thus, an evolutionary information system development approach is utilized. The Domain Construction Strategy is derived from the experiential learning constituent in the Activity Domain Theory.
THE STRUCTURE OF THE FRAMEWORK

In Figure 24 the elements of the Framework are illustrated.

Figure 24. The elements of the Framework

Together these elements are related as illustrated in the conceptual map in Figure 25:

Figure 25. Conceptual map of the Framework and its elements

In the following sections, these elements are discussed in detail.
5.1.1 The Conceptual Model

Besides signifying the orientational structure of the coordination domain, the Conceptual Model also delimits the context of the coordination domain. Items which are not included in Conceptual Model are considered to be insignificant by the actors. Again, it is important to remember that the structure and scope of the coordination domain are a matter of agreement among the actors.

The Conceptual Model serves two purposes. First, it is a shared sign for the actors in the coordination domain, and second, it specifies what is to be implemented in the Information System. The first aspect is usually the most difficult to achieve. The model nomenclature used in Conceptual Model may vary. However, the most important issue in deciding nomenclature is understanding. The actors must be able to interpret the model as easily as possible. To this end an object modelling technique such as for example the Object Modelling Technique (Rumbaugh et al., 1991) may be utilized. This also has the advantage that the model is easily implemented if the Information Systems is an object oriented information system, which is the case with eMatrix (see Section 5.1.4 below).

5.1.2 The Process Model

There are several possible classes of models that can be used as the Process Model. The most important criterion for evaluating them for use in the Framework is their precision in signifying the dependencies between activities. From Section 4.1.5 Temporality we know that the activities are interrelated to the coordination items. Thus it is important that the nature of this relationship is understandable for the actors.

In Figure 26 some process models are illustrated.

The most commonly used class of model is a), in which the activities are in focus while the items are placed in the background. This class of process models is well suited to model situations where the dependencies between the activities are mainly serial. It is less suitable to
model parallel activities. The items are scattered in the model and the item state progression is hard to follow. For example, the item A in Figure 26 a) appears in two disjointed places in the model. Moreover, the progress control in this class of models is usually done by changing the state of the activities, for example ‘waiting for input’, ‘executing’, ‘ready’, etc., and not on the item state (S0, S1, S2 and S3 in the figure). A final drawback of this model is that it is hard to see the interaction pattern between items and activities.

The interaction pattern is in focus for another class of process models: the CRUD matrix shown in Figure 26 b) (see for example Politano, 2001). The CRUD matrix captures which activities impact which items and the nature of that impact (Create, Read, Update, Delete). The dependencies between the items and the activities are more evident in the CRUD matrix than in the activity based models. However, there is no indication of the item states which means that the interaction pattern in the CRUD matrix lacks a vital aspect. Moreover, the matrix form makes the CRUD matrices hard to read, that is the signifying precision of the signifier (the CRUD matrix) is low for the actors.

The final class of process models considered is the Information Flow Diagrams (IFD). This model has three elements: a set of items, a set of activities and a diagram which shows how they interact. The items are represented by horizontal lines which may be conceived as a projection of its related Conceptual Model where the actual relationships between the items are suppressed. The activities are shown at the bottom. The vertical arrows in the diagram pointing downwards indicate input to an activity, whereas a vertical arrow pointing upwards indicates an output from an activity. Wherever an upward pointing arrow meets a horizontal line, the corresponding item changes its state. This means that the diagram in the IFD model shows the dependencies between the activities, which directly corresponds to the definition of coordination given by Malone & Crowston (1994, p. 90).

The Information Flow Diagrams belong to a specific class of process models called entity based models (Humphrey et al., 1989). The IFDs appeared in the Ericsson practice in the early 1990s and their origin is hard to track down. The form used in the IFD is similar in nature to models used in differential analysis using analog computers. Here, the variables were drawn as lines with integrators, corresponding to the activities, operated (see Irwin, 2002). Entity based models are suitable in dynamic development situations where the state progression of the coordination items is reasonably stable and activities and tools are used in a rather ad-hoc manner to produce the result.

In the Framework the IFD class of models is used. Again, a major reason for this choice is the transparency of the model with respect to the actors’ shared understanding. The signifi-
ing precision in this type of models is superior to the other classes of models. In Figure 27 an example of an IFD for the Change Request Process is shown.

![Figure 27. A Change Request Process Model in the IFD notation](image)

A more complicated IFD is shown in Figure 36, p. 124. This diagram shows a process for designing multi-chip modules.

5.1.3 The Transition Model

In order to clarify the meaning of the Transition Model, let’s assume that the we have one activity domain $C$ which provides some outcome $O$ from some initial base $I$. Furthermore, this is done in collaboration with two other domains, $A$ and $B$ (see Figure 28). The environment of $C$ may be considered as an activity domain, the constitution of which in general differs from $C$. This means that the outcome $O$ is conceived differently depending on in which domain it appears, which in turn implies that a transformation between relevant elements inside and outside $C$ must be done.

Furthermore, as illustrated in Figure 28, the outcomes of two collaborating domains $A$ and $B$ must be coordinated in order to produce the outcome $O$. For these domains, $C$ represents the environment, and a similar transition must be done between the elements in $C$ and the interior elements in $A$ and $B$. Thus we see that the same pattern repeating itself no matter which domains are interacting.

In general, the transition between two domains may impact any type of elements like signs, language, norms, etc. For the purpose of coordination we shall concentrate on the management of the dependencies between activities. Two aspects need to be considered: how the
THE STRUCTURE OF THE FRAMEWORK

input to and output from an activity is transformed, and how to capture the recurrent pattern discussed above.

To this end we will use the Specification Based Data Model (SBDM) suggested by Gandhi & Robertson (1992, 1995). This model has basically two constructs: ‘specification’ and ‘implementation’. These are connected by two relations: ‘specification’ is implemented by ‘implementation’ and ‘implementation’ needs ‘specification’ (see Figure 29). Thus we see that recurrence is an inherent feature in SBDM.

For activity domains, SBDM can be interpreted in the following way: The ‘specification’ captures the ‘intrusion’ of an activity domain in another domain, for example, how C appears in its environment. This is equivalent to saying that C is conceived as a ‘black box’ in the environment, where the characteristics of the ‘box’ C is captured by the ‘specification’ entity in SBDM.

If we consider a particular entity in an input or output to an activity, two elements must be transformed: the description of the entity and the state of the entity. In SBDM this element is related to the ‘specification’ construct. An example from Ericsson of description transformation is that products are identified by a name and a revision. When managing the developing a software code belonging to a particular product, this is done in another activity domain where a software configuration tool is used. This tool uses ‘branches’ and ‘nodes’ to identify a particular version of that code. In this case a transformation between (name, revision) and (branch, node) must be done. Similarly, the way of expressing states in the software configuration tool must be transformed into the states of the product.

The other aspect of domain transition concerns recurrence. In SBDM this is achieved by the ‘implementation’ construct which relates specifications through the two relations ‘implemented by’ and ‘needs’. For coordination purposes we may interpret ‘implementation’ as the coordination of the contributions of collaborating activity domains to the outcome. In the Framework the coordination is modelled by the Process Model. Thus the ‘implementation’ entity in SBDM is exactly this process model.

This means that the Transition Model can be illustrated as in Figure 30. The domain specification holds descriptions of the input and output entities, their states and how these are transformed between the inner and outer of the domain. The transformation itself may be expressed in various ways, for example as rules or functions. The domain specification is implemented by collaborating domains.
which are coordinated according to the Process Model. The needed collaborating domains are in turn specified by domain specifications.

This way of modelling the collaboration between activity domains has some nice features. It provides traceability across any pattern of collaborating domains regardless of size and level of the domains. Furthermore, it provides a freedom for different activity domains to evolve their own structure in terms of language, norms, standards, traditions, etc., without relapsing into isolated islands. Finally, it is in line with the well-known principle of separation of concerns in order to cope with complex systems.

5.1.4 The Information System

The purpose of the Information System (IS) is to support the coordination task in the coordination domain. This means that the Conceptual Model, the Process Model, the Transition Model and the Stabilizing Core must all be possible to implement in the Information System. A consequence of this stance is that there is only one information system supporting the coordination task in the Framework. Furthermore, the IS must interact with other ISs in collaborating activity domains. This means that the transformations modelled in the Transition Model also must be implemented in the Information System as well as in the interfacing ISs. These issues are discussed further in Section 7.4 Information System Architectures (p. 186).

Moreover, it shall be possible to use the Information System as an instrument in the Domain Construction Strategy (see Section 5.1.6), which implies that it must be very easy and simple to modify the implementation of the models. In addition to these requirements, other more traditional requirements on the Information System such as performance, global distribution, data security, web-based access, etc., must be considered. Altogether, this puts very stringent requirements on the properties of the Information System in the Framework.

Since the Information System is derived from the tool usage constituent in an activity domain, it is the structure and evolution of the usage domain that drives the evolution of the Information System. In this evolution all elements of the Framework as well as their interactions must be considered.

When conceived as a sign, the Information System signifies important phenomena in the usage domain for the actors. Thus, the signifying properties of the Information System are important. One example is that signifiers such as icons should be the same in the models and in the Information System to avoid the burden for actors to learn several signifiers signifying the same phenomena.

In the same way as it is possible to use different nomenclatures for the Framework models, it is possible to use any IS as long as it complies with the requirements stated above. In the Ericsson practice we have been using eMatrix from Matrix-One Inc., which is an object oriented Product Data Management (PDM) system. This IS has all the desired properties,
THE STRUCTURE OF THE FRAMEWORK

above all the ease of changing the implementation. In Figure 31 the implementation procedure is sketched.

Each signified item in the Conceptual Model a) is implemented as a type in eMatrix b). This includes attributes, names, operations, etc., associated with the item. Dependencies between signified item in the Conceptual Model are implemented as relations in eMatrix c). Also, relation cardinalities (how many items may be related on each side of the relation) and revision stepping rules (what will happen when a new revision is created of an item) are defined, as well as possible attributes on the relations.

The Process Model is implemented as state chains, which are called policies in eMatrix d). A policy is the equivalent to a line the information flow diagram in the Process model. The policy in Figure 31 corresponds to the Process Model shown in Figure 27, p. 110.

Different mechanisms may be used to implement the Transition Model. For example, rules may be used to transform the state values in one state chain to the values in another. Views may be defined which are focused on a particular context. In the actual implementations described in Chapter 6 The history of the Framework the most salient expression of the Transition Model is the so called anatomy (see Section 7.3.3 Orientation, p. 165). An example of an anatomy is the work package structure illustrated in Figure 2, p. 3.

The implementation of the models is mainly done directly in the user interface of eMatrix1. This means that a fairly simple set of models, like the ones shown in Figure 31, is implemented in a matter of minutes. Moreover, the implementation is done without halting the database which means that the implementation may be changed on-line, for example in an

Figure 31. The implementation of the models in the Information System.

---

1. More precisely, it is done in the Business Modeller in eMatrix. Advanced applications are programmed in programming languages, including an interpretative one (TCL).
ongoing meeting. This is the key to the experiential learning way of working in the Framework. If the implementation would take a longer time, say half an hour for changing an attribute, and the database had to be closed down, the experiential way of working would not be possible.

5.1.5 The Stabilizing Core

The stabilizing core is a generic term for elements in an activity domain which have a trans-individual character, that is, they exist only through the activity of individuals but are independent of particular individuals. Such elements may include forms, rules, procedures, conventions, strategies, norms and technologies. It also includes structures of beliefs, cultural issues, shared meanings, etc. These issues have been thoroughly treated in the literature, see for example Lewitt & March (1988) or Larsson (1990).

The reasons why the Stabilizing Core is included in the Framework are given in Section 4.1.7 The stabilizing core. A major incentive is that the extent of the core impacts the ability of the usage domain to adapt to changes. As discussed in Section 3.3.7 Stabilizing core, p. 80, this ability is hampered if the core becomes too encompassing or too small. By adjusting the balance between the extremes an optimal ability to adapt to changes may be found.

5.1.6 The Domain Construction Strategy

The purpose of the Domain Construction Strategy is to provide the actors in the coordination domain with a strategy for how to construct it. This strategy consists of an ongoing iteration between reflection and action phases in which the models and the IS are alternatively reflected upon and tried out in practice. This means that we consider the coordination domain as a being in a state of constant evolution. The change may be more or less evident but nevertheless it is always there.

An important aspect of the Domain Construction Strategy is that a shared understanding about the coordination domain emerges as a result of the strategy. To this end the models and the Information System are conceived as signs which shall acquire a shared significance during the application of the strategy. The following measures are taken to achieve this:

- The experiential learning in itself is a fundamental way for humans to create meaning in the context where they live and act (see Chapter 3 The Activity Domain Theory).
- The number of actors in the coordination domain should be kept as small as possible while still comprising all relevant aspect of the coordination domain. For example, if the coordination domain includes requirements as coordination items, a requirement coordinator should be one of the actors. Other actors may be an expert on the IS and a project manager whose task it is to allocate the requirements to project deliveries, etc. This is discussed in more detail in Section 7.3.6 Deployment, p. 173.
- The nomenclature used in the models should be decided from a perspective of understanding rather than formality. A nomenclature aimed at formal specification should be avoided. For example, although widely accepted as a standard for object-oriented modeling, the UML nomenclature is less suited in this context since it usually is not
THE STRUCTURE OF THE FRAMEWORK

familiar to the actors in the coordination domain.

- Signifiers used in the models and the IS, for example icons, should be identical if they signify the same phenomena.

5.2 Summary
The purpose of the Framework is to operationalise the conception of coordination as coordination domains proposed in this study. The coordination domain is signified by a Conceptual model, a Process model, a Transition model and a Stabilizing Core. The models are implemented in an Information System which is used to support the coordination.

The coordination domain is constructed using an experiential learning strategy which means that the models, their implementation in the information system and shared meaning emerge simultaneously in a dialectical manner, see Figure 32. By working with the models and the information system, these acquire significance for the actors; that is, the models and the information system become meaningful for the actors.

![Figure 32. Results from the Domain Construction Strategy](image-url)
PART III - RESULTS FROM THE ERICSSON PRACTICE

In Chapter 6 The History of the Framework the trajectory of the Framework in the Ericsson practice is reconstructed. The time span is from approximately 1990 until the middle of 2002. Important events are described as well as the origins of the Framework and its elements. This chapter provides an answer to the research question 1.

In Chapter 7 Framework Consequences the overall consequences of the intervention of the Framework are derived from the empirical findings. The penetration of the Framework into the Ericsson practice is described in terms of projects impacted and the structure of coordination domains. Consequences for information system architectures and the construction of shared meanings are discussed. This chapter provides an answer to the research question 2.

In Chapter 8 Reinforcement Rods elements contributing to successful outcomes of projects are derived. The relation between the reinforcement rods and the Framework is analyzed. This chapter provides an answer to the research question 3.
6 The history of the Framework

In this chapter I address the research question “How did the Framework evolve in the Ericsson development practice?” In Figure 33 the timeline of the Framework is sketched. Some major events are indicated. As can be seen, the Framework history spans more than 10 years1.

It is inevitable that the selection of events and my interpretation of these are biased and that other actors may have different interpretations. However, my interpretations are as far as possible grounded in Ericsson internal documents, mails, personal notes, etc.

The history of the Framework is outlined along two intertwined threads. The first thread is that of the Framework itself. Which are the sources of the Framework? How were its parts shaped? In which order did they appear? Which were the major turning points in its evolution? etc. The other thread is the evolution of areas within Ericsson which were outside the immediate context of the Framework but still impacted its trajectory. Such impacts may be corporate directives, organizational changes, changes outside the company, etc.

The history can be divided into a number of fairly distinct phases. These phases are treated in a section each in the following. In Section 6.8 Analysis of the Framework trajectory a concise summary and analysis of the history are given in which constructs from the Actor Network Theory of Latour (1998) are utilized as focusing lenses. This section can be read stand-alone for readers not interested in the specific details.

6.1 A pattern emerges (1990-1995)

For a long time the Ericsson company has been structured around the AXE10 telephone system which became an immense success in the Ericsson history (Vedin, 1992). Late 1990 I started to work in one of the largest projects ever at Ericsson, the purpose of which was to

---

1. ‘eMatrix’ in Figure 33 is a commercial Product Data Management system that eventually became the Information System in the Framework (see WEB-1 for more details).
CHAPTER 6

develop a successor to the AXE10 system. I will refer to this system as the AXE-S (sucessor) and the project as the AXE-S project (the successor project). My task was to work with methods and tools for supporting the design of hardware systems.

The AXE-S project was closed down late 1995 without having reached its goals. However, a lot of experiences from this project were used in subsequent tasks. Many of the ideas implemented in the Framework come from this period. I do not claim to be first to come up with these ideas. On the contrary, taken one by one these are well known. I claim however that my contribution is to have selected and organized these ideas into a coherent theory from which the Framework originates.

The so called coordination constituents of this theory are described in Chapter 3 The Activity Domain Theory. The traces back to the AXE-S project are as follows:

- **Shared meaning**, which concerned concepts, interpretations, understanding, etc. This was manifested in the Framework as the Domain Construction Strategy.
- **Experiential learning**, which concerned system development issues. This was also manifested as the Domain Construction Strategy in the Framework.
- **Orientation**, which concerned information modelling issues. This was manifested in the Framework as the Conceptual Model.
- **Temporality**, which concerned process modelling issues. This was manifested in the Framework as the Process Model.
- **Contextuality**, which concerned issues like organization, cooperation, coordination, context separation, etc. This was manifested in the Framework as the Activity Domain notion and as the Conceptual Model.
- **Domain transition**, which concerned the transitions between activity domains. This was manifested in the Framework as the Transition Model.
- **Stabilizing core**, which concerned issues regarding rules, norms, standards, etc. This was manifested in the Framework as the Stabilizing Core.
- **Tool usage**, which concerned information systems (IS) issues. This was manifested in the Framework as the Information System.

In the following account for the history of the Framework I will mainly describe how the different elements of the Framework appeared in the Ericsson practice. Wherever there is a trace back to the constituents this is indicated as notes in brackets, e.g., [orientation]. In the case quotations are in Swedish I have translated them.

6.1.1 Concept elaboration

It was realized early in the AXE-S project that concepts had to be ‘managed’ in a systematic way. Many new concepts were invented in the project and these concepts had to be relevant, unambiguous and understandable to the actors.

To this end an ambitious sub-project was launched to collect and define the concepts used. The different units had special teams whose purpose it was to collect candidates for new concepts. The pending concepts were taken to a reference group where it was decided if a particular concept would be included in the AXE-S project encyclopedia.
THE HISTORY OF THE FRAMEWORK

In spite of this work, it was very difficult to get an overall picture of the project where all concepts made sense [shared meaning] (LTX-1994-08-15, ERI-1995-03-15). For example, in the new system development process (see Section 6.1.2) more than 120 new concepts were defined in a list without any conceptual map that could explain how they were related to each other (ERI-1993-09-23).

Moreover, towards the end of the project, directives were issued to use the system development process to develop all kinds of products in the project, not only the AXE-S system. For example, the process was used to develop educational material. This meant that concepts like ‘node’, ‘logical reference model’, ‘system entity’, etc., which were originally concepts in the AXE-S system, had to be appropriated into the educational area. This created even more confusion: ‘Is a manual a node or system entity?’ ‘Can a chapter be regarded as a logical reference node?’ and so on.

Thus, in a few years (1990-1995) a completely new organizational language was to emerge, often in conflict with the traditional one. This turned out to be an overwhelming task. In parallel to this a separate, unrelated initiative was started outside the AXE-S project to define one hundred core concepts within Ericsson (ERI-1992-09-30). These experiences indicated that the effort to create a shared meaning in a work setting is in general underestimated, if paid attention to at all [shared meaning].

6.1.2 The system design environments

Many of the ideas for the Framework came from the work with developing a hardware design environment (HDE) in the AXE-S project2. The principles for HDE were laid down in a system description in 1991 (LTX-1991-10-09). The guiding principle was expressed in the following way:

“HDE shall combine a long-lived architecture with a flexible functionality based on purchase”
(LTX-1991-10-09, p. 4)

In this document a modular process architecture is described in which a process core, process components and applied processes are the main elements. The idea was to treat a process like any other product and configure tailor-cut processes from the process components. In this architecture the process core was an early example of a stabilizing core [stabilizing core]. This architecture was further refined in the following years (LTX-1994-07-06, LTX-1994-08-15).

A process component was a package of process descriptions, status checks, rules, tools, templates, etc., with the purpose of providing a designer with a complete set of utilities for a particular design task. The hardware development process was more heterogeneous than the software development process since different types of hardware were being developed such

2. Again, it is important to stress that the many actors, of which I was one, contributed to the results achieved.
as printed circuit boards, ASIC’s, multi-chip modules, etc.\textsuperscript{3}. Thus, the contextual aspect was more salient in hardware design than in software design [contextuality].

Early on it was recognized that the data used in the process components needed to be translated between the interior and exterior of the component. One purpose was to archive data in a format which was independent of the particular tools that might be used in a process component. To this end a mechanism to translate data between process components was developed. This mechanism, called the Design Information Interchange Model (DIM), was defined in the following way:

“DIM is a kind of framework for how to handle interfaces between data. DIM sets up rules for which formats may be used for different views for hardware information, for example behaviour descriptions, logical circuit descriptions, netlists, test data, layout. Moreover, DIM provides tools to simplify translations between formats.” (ERI-1991-08-08, p.2)

The work with DIM was one of the origins for the Transition Model in the Framework [domain transition].

Another source of the Transition Model was the Specification Based Data Model (SBDM) suggested by Gandhi & Robertson (1992, 1995). I came across this model at a conference in Grassau in 1992 where I and a colleague presented a work concerning the co-design of software and hardware (later reported in Östman & Taxén, 1995). The SBDM model directly appealed to me as a potential model for handling the borderline between different contexts:

“The specification-based data model (SBDM) is a unifying framework to model configurations of systems that contain components from differing engineering disciplines.” (Gandhi & Robertson, 1995, p. 336)

My interest in modelling the separation of contexts originated in experiences from hardware design methods. For example, during 1990 I became a project manager for a study with the purpose of investigating the suitability for VHDL\textsuperscript{4} as a design language. One of the key issues was studying translators (LTX-1990-06-18). The purpose of a translator is to make it possible to move between abstraction levels. In (LTX-1990-06-18) an example is given where a transition took place between “integer” and “byte” levels running at different clock speeds. This was one of the first examples of a Transition Model [domain transition].

\textsuperscript{3} ASIC: Application Specific Integrated Circuit.

\textsuperscript{4} VHDL: VHSIC Hardware Description Language. VHSIC: Very High Speed Integrated Circuit
The basic construct of SBDM is a recursive structure of specifications which are implemented by implementations and in turn need other specifications. This was utilized in an attempt to combine the process modules with an SBDM model of the system to be developed (LTX-1994-03-29b). In Figure 34 an example of this is shown where a certain specification can be implemented in either software or hardware using different process components for the design of the implementations [domain transition].

In the document LTX-1994-07-06 the work with the system development process so far is summarized and a number of improvements for the future are suggested. Among these is a contextual view of the architecture; that is, the process model depends on the role of the actors using the process [contextuality].

Early in the AXE-S project a project member suggested the Information Flow Diagrams (IFDs) as a comprehensive way of representing processes, see Figure 35. In LTX-1994-07-06 it is stated that IFDs have a number of advantages. For example, they have a core structure which can be applied to any process module and any layer in a process hierarchy [temporality]. It can also be seen that separation of the process model in a ‘control’ layer and a ‘resource’ layer is compatible with the Malone & Crowston definition of coordination: “[Coordination is] managing the dependencies between activities” (Malone & Crowston, 1994, p. 90).

Another observation in LTX-1994-07-06 is that it is necessary to separate between those aspects of the process that are common to a group of users and what is unique for a particular group. These aspects are called the ‘administrator’ and ‘entrepreneur’ side of the process respectively. Methodology and support systems are considered commodities and as such belonging to the entrepreneur side while rules for identification, quality systems, etc., are more long-lived and belong to the administrator side. Here we can notice a balance aspect which later became manifested as the Stabilizing Core in the Framework [stabilizing core].

The Information Flow Diagrams were later adopted as the standard way of representing processes for hardware design in the AXE-S project (LTX-1994-08-15; ERI-1995-03-18).
One striking example was the development of a multi-chip module, which is a complex integrated circuit. The Information Flow Diagrams for that process component is illustrated in Figure 36 [temporality].

```
Figure 36. Information Flow Diagram: multi-chip module design (ERI-1995-03-18)
```

One observation from the work with the multi-chip module was that an enlarged paper copy of this process model was hanging on the wall in the project room. The progress of the design was marked on the paper for everybody to see. Thus the model served as a communication mechanism in the project which can be apprehended as a composite sign. During the AXE-S project the issue of automating process support was discussed a lot, but it occurred to me that the most important issue was the common view of the process. The automated support did not seem nearly as important, it was quite sufficient to see the paper on the wall [shared meaning].

During 1992 discussions were started to coordinate the software and hardware development processes. The AXE-S system consisted of a large amount of hardware, yet the software process was constantly prioritized over the hardware process. In order to achieve a more even balance a strategy was defined as follows:

"Starting from a common process architecture we first focus on each process area by itself and then we integrate them" (ERI-1992-12-21, p.2)
THE HISTORY OF THE FRAMEWORK

In a report the progress in a number of alignment areas was summarized (LTX-1994-04-01). Only in 7 out of 18 areas did some alignments exist. Most remarkable was that the foundation, the process architecture, was not aligned after more than a year’s discussion. My recollection from participating in the alignment team is that the actors from the software and hardware community could not agree on the basic perspective. The modular process architecture in the hardware process was never accepted in the software community. There were constant discussions about basic concepts and how to organize the alignment work [shared meaning]. The effort to align the software and hardware processes was finally terminated when the hardware development was moved out of the AXE-S project in early 1995.

The experiences from this undertaking triggered many reflections which were later manifested in the Framework. One reflection was that we tried to align too much. A common understanding cannot encompass everything. There must be a balance between what should be coordinated and what shouldn’t [stabilizing core]. The separate practices of software and hardware design should be regarded as cooperating practices, not as one single practice [contextuality]. Finally, this experience again indicated that the effort to agree on the meaning of concepts is vastly underestimated [shared meaning].

6.1.3 Information management

In the report ERI-1989-04-03 general principles for handling of the AXE-S system and the support system in the AXE-S project were defined. In this report fundamental characteristics regarding ‘surviving systems’ are discussed such as stability, flexibility, integration of new components, and autonomy of different system parts. The importance of stability is expressed in the following way:

“A stable foundation is necessary. For humans this is provided by the stable structure of the DNA, which is then found at all levels in the human body. In an organization the business processes are adaptable while the type of information managed in general is indifferent... [the type of information] is structured in information models [...] which make up the stable structure” (ERI-1989-04-03, p. 3)

In the marginal I made a comment regarding the stability of the information which indicate that I questioned the stability of the information models. This is one of the first traces indicating the Stabilizing Core in the Framework [stabilizing core].

In another report, ERI-1991-04-03, the principles for information management in the AXE-S project were defined. The report proposes a way of managing information which is a radical break with the traditional AXE10 world. For example, ‘documents’ are to be replaced with ‘information elements’ as management items. The report also includes a suggestion for an information model for the management. However, there is no discussion about how to arrive at a shared meaning concerning this model. This is simply not an issue [shared meaning].
6.1.4 Information systems

The information system (IS) in the AXE-S project had very ambitious goals from the start. The perspective was to develop a first class system which was superior to any commercial system available on the market. The main principle was that all the information needed was to be managed in one IS only (ERI-1991-04-03).

A forerunner to the IS was being developed during the early years of the project. This development was however terminated in December 1992 on the very same day it was supposed to be released. The reason was that it did not match the requirements for sharp usage in connection with the development of early test nets of the AXE-S system. An interim solution was launched with the main purpose of securing the storage and retrieval of files. Gradually the ambition level was lowered and in December 1993 the internal development of the main IS was terminated. The strategy from that point on was to use a commercial system for software configuration management. This evolution was summarized in 1994 by one of the participants as follows:

“The interim IS was and remains a quick and dirty solution. It cannot possibly evolve into an information management system worth the name. The interim IS was a conscious sub-optimization when it was introduced. It brought the information management ten years back in time. The interim IS is becoming a part of the problem instead of the solution... Now a commercial system for configuration management is introduced. The intention is good. But I see an imminent risk that this will be a continued bottom-up development which makes the ‘temporary’ conceptual world of the interim solution permanent.” (ERI-1994-04-26)

During this period it became evident that the original idea of managing every piece of information in one system was unrealistic. The configuration needs for developing software were quite different from that of managing for example product structures. This pointed towards the notion of activity domains [contextuality]. Different activities simply need different tools. This may sound very sensible and straight-forward, but during this period there was no consensus about this.

In 1993 I became responsible for the technical area called Hardware Support System in the AXE-S project. The need for various information systems tailored to specific tasks was obvious in the hardware community. Also, there were a number of commercial support systems available. This sparked an investigation of commercial Product Data Management (PDM) systems which might serve as the backbone tying all the specific IS together. The investigation lead eventually to the establishment of a Corporate standard PDM system in December 1994. I will refer to this system as C-PDM\(^5\) in the following. Later on this decision became a source of conflicts since the IS in the Framework (eMatrix) was considered to violate the Corporate policy concerning PDM systems.

The final step in the saga of the ISs in the AXE-S project was taken in July 1995 (about half a year before the project was terminated) when the basic principle concerning the IS was

---

5. This PDM system should not be mixed up with the legacy PDM systems that have been used for a long time at Ericsson for archiving products and documents (called PRIM and GASK within Ericsson).
redefined as “The information will always be managed in several systems” (ERI-1995-07-06, p. 5). Thus, the original principle was completely reversed.

No strict analysis was ever made of this mishap. However, a number of issues were brought up:

- The conceptual world of the main IS was never demonstrated and tested. Neither were the theories behind the system (ERI-1994-04-26) [experiential learning].
- It was not possible to understand the system without confirmation in practice. Therefore, the development must be made in steps. (ERI-1994-04-26). Prototyping should have been included in the project from the beginning (ERI-1994-04-26). This experience points towards the experiential learning way of working and incremental development [experiential learning].
- The long term solution was too long term and the short term solution too short. The world of interim IS became more or less permanented (ERI-1994-04-26) [experiential learning].
- One IS cannot be used for wildly different activities (ERI-1994-06-02). There was an idle wish that it would be possible to create one support environment that everybody could use (ERI-1995-12-12). Each activity should have ISs which suits the tasks of that activity [contextuality].
- The handling issues were largely neglected by the project management (ERI-1994-08-15b)
- Lack of a common language (ERI-1994-08-15b) [stabilizing core].
- Difficult to see the whole picture, nobody had control of this, deep isolated islands of knowledge without communication in between (ERI-1994-08-15b) [orientation, shared meaning].
- Requirements were unclear, no sharp customer (ERI-1994-08-15b).
- There was a lack of governing directives. Standards were not followed. We invented our own standards (ERI-1995-12-12) [stabilizing core].
- The bureaucracy in the project was terrible. The development should have been headed by experienced senior designers rather than line managers (ERI-1994-04-26).

6.1.5 Towards the Framework

In 1995, about half a year before the AXE-S project was terminated, I wrote a report which summarized the experiences from working with the design environment for hardware in the AXE-S project (LTX-1995-04-26). This report contains several examples and suggestions which later became manifested as the coordination constituents in the Activity Domain Theory: intersubjectivity, contextuality, domain transition, experiential learning, orientation, temporality, stabilizing core and tool usage. The dialectical approach to system design was reported in Taxén (1995)⁶. Much of the continued work towards the Activity Domain Theory and the Framework is grounded in these two reports. Early 1996 I described the main
features of the Framework in an internal report: “A Coherent Framework for Development” (LTX-1996-02-23). This was the first time the concept of Framework was mentioned.

6.2 Coordinating incremental development (1996-1997)

For some years other projects at Ericsson had started to change their way of working towards an incremental development model for developing large software systems. There were several reasons for this: better management of volatile requirements, avoiding all-in-one integration and testing of the system towards the end of the project, earlier feedback to customers by delivering parts of the functionality earlier, etc., (ERI-1996-04-04). The Framework evolution between the years 1996 - 1997 was mainly related to support for coordinating incremental development projects.

6.2.1 The Incremental Development Method Package

It turned out that projects were using different variants of incremental development and there was a need to consolidate their experiences. Early 1996 I joined a project which should compile the best practices into an incremental development method package (IDMP).

By that time, discussions had already been going on for quite some time about what incremental development was all about. It was difficult to agree on a common meaning of this concept and the discussions did not converge [shared meaning]. In February 1996, I suggested using a conceptual model as a way to overcome this stalemate. The intention was merely to have a picture to which you could point and say something like: “I want this relation to go from here to here instead” or “This object is not important to incremental development” and so on.

With the help of the conceptual model the discussions slowly began to converge [orientation, shared meaning]. In March 1996 the first version of the model was ready. This was further refined into the one illustrated in Figure 37. The boxes signify things that were considered important to incremental development and the lines signify relations between them. Some parts of the model were already existing in the traditional way of working (a waterfall oriented model), but some parts were new. Since the coordination items at this time were mainly documents, most boxes represent different document types. It can also be

6. The AXE-S project was closed down while I was at this conference in Austin in December 1995.
noted that the Specification Based Data Model is included (the ‘specification’ - ‘implementation’ part in the lower right corner).

The IDMP was refined into a product containing guidelines, method descriptions, document templates, etc. Several Ericsson internal seminars were held in Sweden and Holland during 1996 and 1997. In January 1997 the first product release of the IDMP was made (ERI-1996-12-02). This release also contained an early version of the Process Model.

All throughout this period discussions continued about the meaning and interpretation of the models [shared meaning]. One of these discussions concerned how to modify the standard Ericsson project management method PROPS\(^7\) for incremental development. Here is a discussion from November 1996:

“During our work with incremental development and PROPS, the generic version, we have come across ambiguities in the definitions in the UAB\(^8\) method package. The basic inconsequence lies in the fact the activity and the result of an activity are confused by giving them a common name”

(PROPS developer, ERI-1996-11-05)

“To me it is obvious that ‘project’ stands for both result and activity. How we separate these can best be clarified in a conceptual model. All in all, the model shows how we thought when we integrated incremental development into existing methods. We can’t describe the world any better

---

7. There seems to be no official explanation what this acronym stands for. The terminology guide to PROPS merely states that “PROPS is the name of the Ericsson project management method.” (ERI-1994-01-01)

8. UAB: Ericsson Utvecklings AB, the Ericsson unit where I worked at this time.
CHAPTER 6

than anyone else can. But we are pretty good at describing how we think about it (IDMP project manager, ERI-1996-11-05)

The method package was used in a sharp project called CMS-30 phase 7 which had a Japanese operator as a customer. The conclusions from the evaluation of this project were a.o.t. (ERI-1997-10-23):

• Incremental Design is a possibility, but needs some improvements.
• Milestones are to be planned per increment (needs a good tool).
• It must be agreed that we need several TG2’s for one project9.

By that time a good tool was on its way in at Ericsson [tool usage].

6.2.2 The Information System enters the stage

Already in January 1996 the tool issue was discussed and a requirement specification was written in March 1996. It was clear that none of the existing tools did fulfil the specification. In May 1996 I was visiting a construction fair in Gothenburg when I by coincidence came across a lonely looking salesman at a terminal. Roughly the following conversation took place (checked with the salesman):

- LT: What have you got here?
- SM: It’s a tool called eMatrix, I’ll show you.
- LT: Can you implement this model? (I drew a sketch of the Specification Based Data Model, see Figure 29, p. 111).
- SM: Sure, no problem (a couple of minutes later he had implemented the model and instantiated some objects from it).
- LT: Wow! That was impressing!

This unlikely event led to further contacts which eventually established eMatrix as the Information System in the Framework. eMatrix was originally a Product Data Management (PDM) system aimed at industrial management of globally distributed, large quantities of product data and with many thousands of users. In September 1996 eMatrix was suggested as the tool for supporting incremental development and in October 1996 a demonstration license was bought by Ericsson. The first installation of eMatrix was done in April 1997. The next step was to start up some pilot projects.

6.2.3 The pilot projects

In May 1997 a consultancy company was engaged to contact potential pilot projects within Ericsson. A tool prototype called CPLtool (Construction PLanning tool) based on eMatrix was developed. The first demonstration was held on June the 9th, 1997. Later that month

9. TG2: Tollgate 2, a decision point in PROPS where a decision is taken to go ahead or not with the execution of the project.
contracts were written with pilots in the Netherlands, Germany and Karlskrona. I became the technical project manager for the development of the CPL tool.

During this period the Conceptual Model evolved into the one shown in Figure 38.

Each box in the model was directly implemented as a type in eMatrix and each line as a relation. In addition to this, state chains, attributes, cardinalities and revision stepping rules were implemented. The implementation was tried out in practice and modified whenever needed. In comparison with the model in Figure 37, the tool supported model shows less document oriented types. For example, the so called Master Configuration Index (MCI) visible in the first model is now replaced by relations. The boxes at the bottom (ANT, CNT, etc.) denote Ericsson specific product and document types.

The model also contains types that are related to project management, for example Resources, Increment Task Specification, etc. The box ‘Impact’ is in fact an attribute on the relation between Resource and Feature Increment which holds estimates for the effort to develop a certain increment.

Thus, it is can be seen that the introduction of the tool opens up new possibilities to construct the coordination practice. Furthermore, the Conceptual Model, the Process Model and the implementation in the Information Systems were continuously changed as the construction of the domain evolved [experiential learning].
In Figure 39 an example of a view from eMatrix is shown where instances of the types defined in the models can be seen along with their relations.

<table>
<thead>
<tr>
<th>ObjectType</th>
<th>RelationType</th>
<th>RelationValue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer Napier Telecom Inc</td>
<td>CustomerToCustomerFeature</td>
<td>CustomerToCustomerFeature</td>
</tr>
<tr>
<td>Customer Feature Flexible Charging</td>
<td>CustomerToCustomerFeature</td>
<td>CustomerToCustomerFeature</td>
</tr>
<tr>
<td>Customer Feature Statistics Improvements</td>
<td>CustomerToCustomerFeature</td>
<td>CustomerToCustomerFeature</td>
</tr>
<tr>
<td>Customer Feature Software Supervision</td>
<td>CustomerToCustomerFeature</td>
<td>CustomerToCustomerFeature</td>
</tr>
<tr>
<td>Customer Feature TRAF</td>
<td>CustomerToCustomerFeature</td>
<td>CustomerToCustomerFeature</td>
</tr>
<tr>
<td>Set of Requirements 239/7056-FCO 103 200 C</td>
<td>CustomerFeatureToSoftwareRequirement</td>
<td>CustomerFeatureToSoftwareRequirement</td>
</tr>
<tr>
<td>Feature Increment TRAF</td>
<td>SoftwareRequirementToFeatureIncrement</td>
<td>SoftwareRequirementToFeatureIncrement</td>
</tr>
<tr>
<td>Implementation Proposal 155-01 239/7056-FCO 103 200 C</td>
<td>FeatureIncrementToFeatureProposal</td>
<td>FeatureIncrementToFeatureProposal</td>
</tr>
<tr>
<td>CHT 239 2000 R1</td>
<td>FeatureIncrementToDesignItem</td>
<td>FeatureIncrementToDesignItem</td>
</tr>
<tr>
<td>CAE 137 3482 R1A</td>
<td>ConsistsOf</td>
<td>ConsistsOf</td>
</tr>
<tr>
<td>Source Program Information 135 55 150 55-CMA 107 5472 A</td>
<td>DescribesBy</td>
<td>DescribesBy</td>
</tr>
<tr>
<td>Source Survey 135 14 155 14-CMA 107 5472 A</td>
<td>DescribesBy</td>
<td>DescribesBy</td>
</tr>
<tr>
<td>Source Parameter List 135 13 130 72-CMA 107 5472 A</td>
<td>DescribesBy</td>
<td>DescribesBy</td>
</tr>
</tbody>
</table>

Figure 39. The Information System in the Framework, example (1997)

In addition to the Conceptual Model and the Information System a Process Model was defined using the Information Flow Diagram model, see Figure 40.

Figure 40. The Process Model in the Framework, example (1997)
Thus during this time Conceptual Models, Process Models and Information System implementations evolved in the pilot projects. The models and their implementation were however different in all pilot projects, indicating the need to adapt these to local circumstances [contextuality].

The results of the pilot project were in general positive. Below is a statement from the pilot in the Netherlands:

“To summarize the reactions: very positive!... General comments were: “It is very flexible”, “That tool would be very helpful”, “I’m really impressed and had not expected such a advanced tool”. (ERI-1997-10-20)

Similar reactions came from the pilot in Karlskrona in a report from one of the consultants:

“The demo this time went excellent! What the participants mentioned and I also noticed was that implementing their own project into the prototype made a BIG difference. They were able to see ALL of their design base, ALL of their increments, ALL of their impacts for each of their increments, a very detail and accurate Increment Impact Matrix, some Milestones definitions, and more... (AND ALL THESE WITH ONE TO TWO DAYS OF WORK). “ (ERI-1997-11-05)

During the autumn 1997 the existence of the CPLtool became more widespread at Ericsson. In October a demonstration was held for a Corporate IT group working with IS architectures. The group showed interest in the tool. However, Corporate IT had already decided to recommend the C-PDM system. In November 1997 the system manager for the Methods & Tools unit at UAB decided that the C-PDM system should be used as the tool supporting incremental development (ERI-1997-11-10). Since it was much harder to change the implementation in the C-PDM system this decision, if enforced, would have made the experiential way of working utterly awkward if not point-blank impossible10 [experiential learning].

I argued against the decision to replace eMatrix with the C-PDM system. It was decided that further investigations were needed in order to clarify the requirements on the production version of the tool, particularly the need for flexibility and adaptability. In December 1997 the steering group for Methods & Tools unit decided to put the work with CPLtool in a pending state until more information had been collected from the users in the pilot projects.

At the end of 1997 the demonstration license that the pilot projects had been running on would cease to be valid. This meant that the pilots would have to be stopped immediately. Since the steering group was not prepared to enforce this, a decision was taken on the 8th of January 1998 to buy the first commercial set of licenses for eMatrix. This was one of the turning points in the evolution of the Framework at Ericsson.

In December 1997 I sent the following e-mail to the participants in the CPLtool project:

10. For example, adding a new attribute in C-PDM implies closing the data base and re-compiling the system, something which is not needed in eMatrix. What is done in a matter of minutes in eMatrix is a matter of hours in the C-PDM system.
“Finally, I would like to thank you all for your outstanding work this year. I firmly believe that this is the forerunner of the next leap in the methods and support area for AXE development, and it is not easy to be up front.” (ERI-1997-12-17)

The last words of this statement would become distressingly clear in the period that followed.

### 6.3 Termination and despair (1998)

For some time Ericsson had been negotiating with the Rational software company (WEB-2) to take the full responsibility for software development methods and tools at Ericsson. An agreement about this was signed in December 1997. The intention was that Ericsson should concentrate on its core business which did not include developing methods and tools. Consequently, the Methods & Tools unit at UAB was shut down in late 1998.

Rational claimed that the IDMP was not needed since incremental development was included in their portfolio. No analysis was however made of the similarities and differences between the approaches. The maintenance of the IDMP was transferred to the Netherlands in January 1998 where it eventually ceased to play a role in the organization.

Moreover, Rational stated that eMatrix could be replaced with their own tools. This meant that it became virtually impossible to get funding for the continued work with the pilots. During the first half of 1998 I made a number of demonstrations of the CPL tool in order to engage allies from wherever (see Table 11, p. 42). This period can most properly be characterized as a ‘peddling’ one where utmost despair was mixed with a conviction that the Framework initiative was on the brink of being terminated. One consequence of this was that I started my Ph.D. studies in February 1998 in order to ‘save’ the Framework ideas and results in some way11.

In the meantime a major project, which I will call the ICH project had been launched in Canada to appropriate the methods and tools of Rational for the AXE10 development. By coincidence, I happened to meet one of the project managers of ICH in the printer room and I showed him some tool snapshots. He became very exited. “This is what we need!”, “Has Rational seen this?”, “Why haven't they told us about this?”, “Can you do this and that?” and so on (reported in an e-mail, 1998-04-24). Then he asked me if I could go to Canada next week...

This event resulted in a workshop in Canada in June 1998, the objective of which was to determine the suitability of eMatrix in the ICH project (E-mail 1998-05-22). Two of the consultants working with the CPL tool participated in the workshop. However, the result was a complete breakdown. The expectations were quite different, as somewhat sarcastically related in the consultants’ report:

> “The purpose of the workshop was reviewed together with S and R. Our understanding of the task was to make adjustments of an existing Ericsson eMatrix model to make it fit the ICH environment

11. This research task was made worse by a decision in March 1998 to purge the entire document archive from the AXE-S project.
THE HISTORY OF THE FRAMEWORK

and thereby create a clearer understanding, from a technical standpoint, whether eMatrix can be used as a complement to the already defined ICH tools. A proof of concept exercise. Mr. S’s understanding was that we during the 3 days should produce a system ready for use in production. Mr. R wanted to discuss licensing policies and pricing.” (ERI-1998-06-26, p. 1)

Meanwhile, the pilots managed to continue by allocating resources on their own. The following statement came from the pilot in Karlskrona:

“By the end of this week we'll have all their IP's [Implementation Proposal] entered and the tool will be ready for real use in the project! They are very excited and they are really committed to the project. Jorgen already entered the increments and allocated them into ADs [Amendment Directive]. The more we use it the more we come with innovative ideas of how it helps the project. Mailing a whole increment to a team who then expands the increment to view their assigned items is a really COOL and HI-TECH feature. Jorgen and Tommy said that.” 12 (E-mail from the consultant, 1998-02-02).

This statement is nice example of the experiential learning constituent [experiential learning] showing that new knowledge is gained in actual use. The remark about mailing increments refers to the fact that objects in eMatrix can be mailed to a receiver like an ordinary e-mail. Thus a central increment planning could be combined with decentralized increment development by a team.

In July the Karlskrona site decided to continue with the CPLtool. However, this initiative was never firmly established in the organization and was finally terminated when the product line in Karlskrona was closed.

Early 1998 a large and important project in Germany, the so called AXE Mobile Core project (AMC), showed interest in CPLtool. After a long evaluation period AMC decided in October 1998 to become a pilot. However, this initiative went into a hibernating phase during 1999 and was not revived until early 2000 when the UMTS project decided to go for eMatrix (see Section 6.5).

In a report in April 1998 the experiences from the CPLtool so far were summarized by the consultant working in Germany (ERI-1998-04-20, p.13):

- **Prototyping in real projects:** It is essential for getting the right requirements to use the prototype in a real development project. It is possible to get good requirements and suggestions by just giving demos and having interviews with projects, but to evaluate the requirements regarding usefulness and feasibility they have to be tested in a real project under normal circumstances. When the people start to work with the prototype they come up with numerous ideas how to further improve and extend the functionality of the prototype. Only the people later working with the tool can find out the real needs (requirements) and identify potential problems [experiential learning].

- **Commitment from pilot:** It is absolutely necessary that the pilot project commits to the pilot work. They have to invest resources, time and money. They have to provide all project related information, come up with requirements, use, test and evaluate the pro-

Identifying crucial prototyping parts: It is necessary to identify the really crucial parts of the work with the prototype. If these parts are not found or not considered, they can kill the whole pilot work. In our example, the crucial part is to enter the IP/FF data and to keep it always updated. For big projects, it is impossible for a single person not directly involved in the project to get all the data and to enter it. Even for the people in the project, it is very hard and time consuming to keep track of all changes in the data. If the IP/FF data in the prototype is not complete or out of date, then the prototype will produce wrong results. Before starting the pilot work, these killing factors have to be identified and solved, otherwise, they will threaten the pilot success.13

Flexibility: The platform of the prototype has to be extremely flexible, i.e., numerous adaptations of the prototype must be possible (so that they can be implemented, demonstrated, used, and evaluated) and it must be possible to incorporate the changes and new features very fast (time is always crucial in pilot projects). Working with the pilot is not possible when new requirements from the pilot will take 3 months to implement due to time and cost problems. The flexibility is especially important when several pilot projects are started because every project (although in the same corporation) is quite different from each other and needs various adaptations (different usage scenarios, different documents, working procedures, different sizes of the projects,...). It is impossible to develop a nice tool which is suitable for all kinds of projects without adaptations. The eMatrix platform has the needed flexibility.

Consider different views: It is important to talk to numerous different persons of the pilot project because everybody has a different view on the prototype according to his/her role in the project. Very important is the opinion of the people who will work later with the tool (e.g., PM, IP writer, CM)14. Involve as many different people in the prototype work as possible to minimize the possibility of missing important requirements for the tool or to miss blocking factors of the work with the prototype.

Different needs: When selecting pilots, try to get different ones (small and large, AXE10 and non-AXE10, total project and subproject,...) because they have different requirements for the tool.

Funding: The funding of the prototype work should come from two sides: from the people interested in the results of the prototyping (provider of prototype and later builder/sponsor of the productified tool) and from the pilot side. The prototype provider should take the license costs and the costs of adapting and further development of the prototype. The pilot should take the costs to the support and training they get. Furthermore, the pilot has to take their own costs, i.e., the effort the people have to invest for discussions and for using the prototype. One advantage, if the pilot also takes some costs, is that this will enforce their commitment. If the costs

13. IP/FF data: A matrix showing the estimated effort for developing a particular increment.
are not split between the parties then this very often creates problems with the side taking all costs: they don’t see why they have to pay everything although the other party will have benefits.

In these statements many of the constituents of the Activity Domain Theory can be found: contextuality, experiential learning, stabilizing core, domain transition. Furthermore some reinforcement rods can be found such as participation and focus (see Chapter 8 Reinforcement rods).

So, the first half of 1998 was a schizophrenic period. On the one hand, very promising results were achieved in the pilot projects. On the other hand, the whole Framework endeavour was on the brink of termination. This was however soon to change.

6.4 Renewal in conflicts - the pioneering S-domain (late 1998-1999)

One day in May 1998 a person asked me to demonstrate the CPLtool for him. It turned out that he was the project manager for the Beamon project which was developing a new switching equipment for the AXE platform. This project manager, PM2, had tried out various project management tools in previous projects but found them inadequate. After this demonstration PM2 decided to try out the CPLtool in Beamon. This was to become the first, sharp project using the eMatrix at Ericsson15.

6.4.1 The Beamon project

Beamon was a large, globally distributed project with the main development sites in Stockholm, Italy and Australia. The work to adapt CPLtool to the needs in Beamon started during the summer 1998. One focus was to manage engineering change orders in a better way; that is, requests for making controlled changes in the project. At Ericsson these are called Change Requests (CRs). The CRs are closely associated with baselines in the project. When a project has reached a certain stage the state of the product, its describing documents and various other items are frozen and included in one or several baselines. A change to any of the baselined items must be proposed in a CR. This is processed according to an established procedure where the CR is analyzed in detail and approved or rejected by a specific group called the Change Control Board.

Another focus was requirement management. The traditional way of managing requirements at Ericsson was to write them down in a requirement specification (RS) and make new revisions of the RS when the requirements were changed. PM2 wanted to have better control of individual requirements, something that was not possible with the RS document.

This meant that Beamon had quite a different scope than the pilot projects where the support of incremental development was in focus. This was still interesting in Beamon but the management of CRs and requirements were prioritized. In July 1998 I started as a project

15. Much of the continued discussion is focused on eMatrix as the Information System in the Framework since this was the most salient and controversial part of the Framework. However, the other parts of the Framework (in particular the Conceptual Model) was always present.
CHAPTER 6

manager for the adaptation of eMatrix for the Beamon project. During the autumn 1998 extensive prototyping was carried out together with configuration managers, requirement coordinators and PM2.

On October the 30th 1998 a decision was taken in to go ahead with the use of the eMatrix in Beamon. This was however not uncontroversial. The initiative came from a single project manager, that is, it was not endorsed by top management. Moreover, it was against Corporate policies and the agreement with Rational. This is reflected in the protocol from the decision meeting:

“Decision: to use eMatrix in one project, Beamon, and if that is successful to transfer it to other projects. It is important to remember that eMatrix is a ‘gap filler’ until Rational has a corresponding product” (ERI-1998-11-03).

This decision was followed by an extensive preparation and testing period which included issues like

• elaborating the Conceptual Model, the Process model and their implementation,
• importing product data from the Ericsson product archive,
• defining users and access rights in eMatrix,
• checking the infrastructure, network, etc.,
• integrations to document archive systems,
• installing client systems on user terminals,
• training users,
• documentation, etc.

On May the 10th, 1999, the traditional way of managing CRs and requirements was shut down and the eMatrix was in use in a sharp production environment for the first time at Ericsson. On May the 19th the first CR was issued in the new system:

“Yesterday you all received a mail from me (automatically generated from the MATRIX tool) about Change request 109 020 CR-101 PA1. This CR is the first to be handled by the MATRIX tool.” (E-mail from the Configuration Manager in Beamon, 1999-05-10)

In June 1999 the specification of what the eMatrix should achieve in Beamon was fulfilled. This work was carried out with quite a few resources: one assistant, two consultants from the vendor of eMatrix (one of which was VDR1) and myself. This effort is surprisingly small, something that Larry Bowen, an authority on Configuration Management lecturing at Ericsson, expressed in a mail:

“Lars, Yes I have looked at it but only briefly. The one thing that I did notice was that so far your system contains only meta data not the files with the drawings, etc. What a Herculean\(^\text{16}\) job you have done with very limited resources! It’s going to really be difficult for you to do too much with the system with your manpower level. It amazes me that you done so much with so little so far. Keep

---

\(^{16}\) It so happened that the next project to start using eMatrix was called Hercules...
me posted on where Ericsson is going with the system. (E-mail conversation, 1999-07-07, Larry Bowen, Jet Propulsion Laboratory)

Later during 1999 the application in Beamon was amended with support for test management [experiential learning]. This was not in the original specification. All in one the Conceptual Model in late 1999 had evolved into the one in Figure 41. As compared to the models in Figure 37 and Figure 38 this model looks quite different although some reminiscences from the earlier models can still be traced. This indicates that actors in different areas of Ericsson have different meanings about the coordination context [shared meaning]. Moreover, the needs are different in some aspects and common in other [stabilizing core].

![Figure 41. The Conceptual Model (late 1999)](image)

6.4.2 Fighting problems

During the deployment of eMatrix in the Beamon project several problems were encountered. The set-up of the eMatrix system was based on one server in Stockholm serving various types of clients. This meant that the set-up became very dependent on efficient web-technology and a fast network. Moreover, the clients should work in a heterogeneous environment both on PCs and work stations.

Already in February 1999 it was noticed that the web-client in eMatrix was unstable. The so called full client could not be used at all in Australia because of very long response times.

17. An alternative set-up would have been to install servers in Italy and Australia and replicate the data between them. This was however not realistic with the resources and budget that was allocated to the eMatrix project in Beamon. It can be noted that the architecture chosen is in fact a return to the Information Resource Management (IRM) - architecture (Axelsson, 1998).
These problems became a risk in the project as stated by the system manager for Beamon in Italy:

“Hello [...] yesterday we raised some issues [...] concerning matrix tool. I started making some sort of checks towards especially the technical coordinator about the feeling in using this tool: the answer something like: - almost nobody is using the tool. - It seems quite difficult to start using it. - For EPA and TEI there are additional problems due to the remote access to the matrix server, usually quite slow. - Moreover, there are problem using the tool from the web client interface, since some functionality (like get a report from an item, to printout or to save) are not working properly. I had also the confirmation from our LSO in TEI that to work properly, it should be used from PC.”

(E-mail, 1999-07-09)

The problems could be controlled by various work-arounds but the attitude of the users towards the tool in Italy and Australia was quickly becoming hostile. These problems were further escalated in July when the situation in Australia jeopardized the continued use of eMatrix in Beamon:

“It is very important that we can fix this. Otherwise we may have to stop using eMatrix in Beamon and that would be a dire fiasco.”

(PM2, e-mail, 1999-08-06)

One consequence of this was that VDR1 was sent to Australia in August to try to fix these problems. On site he discovered an inconsistent picture; some clients worked fine and other were very slow. The proxy server to the local Australian network also was not working properly. Another issue was that the support on site felt at times very much abandoned by the support group in Stockholm:

“We at EPA have been having very serious reliability/stability problems with MATRIX. I have sent a number of mails to various people to follow up but the response has been slow and not helped when it does come. I know it is summer up north and this means many vacations but we are quite busy here and need better and faster support. [...] There needs to be some improvement and coordination in available MATRIX expertise. As I said I have sent various mails with MATRIX error messages and problems. Now I need some answers.”

(MT2, local eMatrix support in Australia, e-mail 1999-08-04)

The situation was however stabilized and remained so until October 1999 when an error in the web-client was discovered. This error occurred only in the specific combination of Windows 95 and the current eMatrix-release installed at UAB. The error was found also in a new release of eMatrix which caused a five month delay in the transition to a newer release. In February 2000 the vice president of Matrix-One was visiting UAB for ‘serious and frank’ discussions of how to remedy this situation.

The problems with long response times and unstable performance nearly caused a user mutiny in Australia. Here are some examples:

18. EPA, TEI: Ericsson designations of the Australian and Italian site. LSO: Local Support Organization.
THE HISTORY OF THE FRAMEWORK

“You are probably aware of this, but there is a large number of CRs that should be written but have not as yet. There are mainly related to changes made to FDs after they have been reviewed but there are more. The main reason for this is the MATRIX tool. I know that you at UAB have no problem but we at EPA (and I think TEI as well) have nothing but trouble. It is the exception rather than the rule that the tool actually works here at EPA, which causes everyone frustration and hence a backlog of unwritten CRs.” (Sub-project manager, Australian site, e-mail 2000-02-18).

“As you most undoubtedly heard, MATRIX at EPA is a nightmare to use. I couldn't get it to work at all today and was wondering if you could help me out.” (design engineer, e-mail 2000-02-18)

During this period, one of the developers at UAB had begun to investigate a new web-client based on a servlet technique. This turned out to be much faster than the Java-based standard web client. In March 2000 this client could be released together with a new release of the eMatrix tool. This finally solved the problems with long response times and unstable performance:

“Hi, I used this today to enter our comments on the current CRs. It turned what would have taken me more than an hour using the web client into a 20 minute job. Fantastic!” (Sub-project manager, Australian site, e-mail 2000-05-24)

“The HTML applications/wizards technology has shown drastically improvements in speed and could work acceptable for Australia and Italy.” (Beamon meeting, minuets, 2000-04-20)

The nightmare was over. From that point on the performance of eMatrix was not an issue during the study period.

6.4.3 Escalation

During 1999 two more projects became interested in using eMatrix. One project, Uranus, developed a new version of the central processor in the AXE system. The other project, Hercules, developed an exchange terminal to the central switching equipment. Both these projects consisted of both hardware and software. In June 1999 an installation of clients was made in Norway for the exchange terminal project and later on Croatia joined in as a sub-project. In August representatives for these two projects were included in the eMatrix project meetings in Beamon. At this time the environment was not fully developed in Beamon. In addition, the severe performance problems were experienced in Australia. Moreover, the two new projects worked differently from each other and Beamon, which created lengthy discussion about the Conceptual Model and its implementation in the eMatrix tool. For example, Beamon and Hercules could never agree on the state chain of a baseline which resulted in two different state chains being maintained in the tool [shared meaning].

On February the 15th of 2000 Hercules project started to use eMatrix. Thus it became the second project to rely on the eMatrix for their work. It was followed by Uranus and still another project called HANSI.

19. FD: Functional Description, a document specifying how to implement a function.
However, all throughout this period the distrust from the line organizations prevailed. The resource allocation was constantly questioned which made it difficult to hire the necessary consultants. During the autumn 2000 the situation became worse and I decided to move to the Core Network\textsuperscript{20} unit at Ericsson to continue the work with eMatrix there.

6.4.4 Conflicts

When the first prestigious project started to use eMatrix, the conflicts with Corporate intensified. The decentralized organization of Ericsson has the consequence that various support tools are being developed at different Ericsson sites around the world. These tools are in general specific for each site and not coordinated. Although they fulfilled certain needs, the downside is high costs to development and maintenance, overlapping applications, problems with data transfer between tools, etc.

For Ericsson it was of utmost importance to reduce these costs by reducing the number of tools. The strategy was also that only one type of PDM systems should be used throughout Ericsson. From the Corporate point of view eMatrix had no place in this overarching strategy. However, since some projects had started to use eMatrix, these could not be jeopardized. Instead, Corporate sought to prevent the further expansion of eMatrix.

One way of doing this was to obstruct the access to the IS/IT support group at UAB. Without the assistance of this group new releases of eMatrix could not be installed, network problems could not be investigated, backups could not be taken, etc. The IS/IT group was controlled by a steering board. One of the members in this group was K, the program manager of the Rational program at Ericsson\textsuperscript{21}.

At a meeting in February 1999 in the steering board it was decided to support eMatrix only as a prototype tool:

\begin{quote}
"The introduction of the eMatrix application: There is no decision or forecast of broad introduction. K remarked that eMatrix is directly divergent from the upcoming [Corporate] standard. Decision: eMatrix shall be regarded as a pilot. If a broader usage is anticipated this must be forwarded to the Steering Board."
\end{quote}

(ERI-1999-02-15)

Moreover, it was decided that the new projects (Uranus and Hercules) had to await further decisions before they could start using eMatrix (ERI-1999-05-31). The concern from the Corporate point of view is clearly expressed by K in the following e-mail:

\begin{quote}
"Hi, a clarification. I have no objections against letting the projects that chose eMatrix half a year ago get the support they need. As I remember it was Beamon that was discussed, nothing else. They should pilot the usage and someone should support that. That's between them and Z\textsuperscript{22}. However, I
\end{quote}

\textsuperscript{20} This unit had the overall responsibility for the core network in the 3\textsuperscript{rd} generation of mobile systems.

\textsuperscript{21} The program manager was responsible for the contacts with Rational and the deployment of the Rational products within Ericsson.

\textsuperscript{22} The IS/IT group at UAB.
According to Corporate the future solutions should be built on the existing and upcoming Corporate standards. eMatrix was not part of this since it had emerged from ‘the floor’. This principle blocked Corporate from taking part in any analysis about pros and cons concerning the use of eMatrix.

However, this picture was not unambiguous. A member of the Corporate IT group (IT2, see Section 2.2.2 Interviews, p. 36) supported the incremental development method in IDMP and allocated resources to continue the pilot projects, although he did not want to see eMatrix as the IS in the Framework. He also wanted me to contribute to Corporate conceptual modelling initiatives (E-mail, 1999-10-04). This ambiguous stance from Corporate is reflected in the following decision:

“METHOD AND TOOL FOR SUPPORT OF INTEGRATION-DRIVEN DEVELOPMENT AT UAB. This is handled by Lars Taxén at UAB and [Corporate] has given some support to Lars awaiting the start of DTL23 where the question should reside and be considered. DTL declared that the use and support of this method and tool is a UAB question. DTL do not see that supporting the method and the tool is a corporate question.” (ERI-1999-10-21, p.2).

Thus, in 1999 the prime concern of Corporate was to let the current projects continue on their own and prevent the further spread of eMatrix within Ericsson. However, this was in fact already happening.

6.5 Other practices catch on (2000)

During 2000 two more development sites began using eMatrix. The coordination domains being constructed at these sites were independent of each other24 which meant that the Conceptual Models at these sites (and the implementation in eMatrix) evolved separately. Thus in 2000 three independent domains existed at Ericsson (the S-domain in Stockholm, the L-domain in Linköping and the A-domain in Aachen).

6.5.1 The L-domain in Linköping

In October 1999 the Base Station System (BSS) unit at Ericsson became interested in eMatrix. This unit was responsible for developing base stations and base station controllers in the mobile network. At a meeting in November I gave a demonstration of the eMatrix application in the S-domain. After some discussions BSS decided to investigate the consequences of using the eMatrix for their purpose. This report stated that a combination of eMatrix and the Rational tool suite was the preferred alternative (ERI-2000-02-10). The eMatrix imple-
mentation in the S-domain was copied to the L-domain in Linköping and in May 2000 the work to adapt this implementation to the way of working at BSS began.

6.5.2 The A-domain in Aachen

During 1999 the development of the 3rd generation of mobile systems, the so called UMTS25 system, was intensified. Some nodes in the UMTS network are very complex (see Figure 2, p. 3) and it was recognized early by the project management that the traditional methods were inadequate:

"The total technical changes being implemented in this project are enormous. Such changes are needed in order for Ericsson to get a world-leading product first to market. Using traditional methods then the scope of change implemented in single steps will be too large and can not be managed." (Total project manager, ERI-1999-12-01).

The approach was to break the system down into small parts, and build it up from the base to total functionality in a step-by-step manner, verifying system behaviour as each step evolves. This was basically the same approach as in the Incremental Development Method Package (IDMP, see Section 6.2). The terminology was changed and some new ideas introduced. For example, the Feature Increments in the IDMP was now called Work Packages and the Amendment Directives were called Latest System Version (LSV).

Moreover, this was one of the first projects to use the software configuration management tool ClearCase from Rational in order to coordinate the work of about 500 software developers working all over the world. At first, the intention was to manage the integrations of the LSV’s in ClearCase as well, but this soon turned out to be a bad solution:

"... the idea in the very beginning was to do this as well in ClearCase by some sort of browser or viewer on top of it. But what we have seen is that would take a hell of a lot of design work and that there are other tools which can manage this far better. What we said is that we should separate the revision handling part in the Configuration Management from the content handling. That’s basically why we introduced eMatrix, more planning aid." (Interview with PM3, 9:07).

After the shut-down of the activities around CPLtool in Aachen, the contacts were taken up again in October 1999 when I gave a demonstration of the S-domain application. In December 1999 the project manager for the UMTS project ordered the use of eMatrix in the project (ERI-1999-12-01). In January 2000 the adaptation work began and an extensive prototyping period followed. In May the same year, the UMTS started to use eMatrix for controlling the status and tracking of Work Packages, as well as managing the integration of the LSV’s.

---

25. UMTS: Universal Mobile Telecommunications System
The Conceptual Model for the A-domain is illustrated in Figure 42:

As can be seen, this model hardly bears any resemblance to the previous models in Figure 37, p. 129 and Figure 41, p. 139. This reflects both a different focus and a different way of working [orientation, shared meaning, contextuality].

The A-domain was to become the most successful at Ericsson. One of the project managers, PM4, expressed this in the following way:

“... especially for the execution part I think we would not have been able to run this project without the tool. I think if you simply look at the number of work packages, the number of products that we have delivered, the number of deliveries that we have had, if we would have to maintain that manually, that would have been a sheer disaster.” (PM4, 05-1:07).

Thus, as I summarized in the following e-mail, a circle was completed:

“Scene 1: The IDMP is developed in blood, sweat and tears at UAB [1996-1997, including eMatrix]. Scene 2: The Rational agreement kills this initiative since RUP\textsuperscript{26} shall be used instead [1998]. Scene 3: The Work Package concept (in principle the same as IDMP) is developed in Aachen [1999]. [...] Scene 6: UMTS decides to use the Work Package concept [1999]. Scene 7: UMTS decided to use eMatrix to support the Work Package concept [1999].” (E-mail, 1999-12-15)

### 6.6 A central concern - the C-domain (late 2000 - 2001)

During this period eMatrix became increasingly a central concern for Ericsson. A number of meetings were held regarding the future evolution of eMatrix within Ericsson. Corporate still regarded eMatrix as a local initiative. For example, in a report early 2000 about the future positioning of Information Systems within Ericsson, eMatrix is not mentioned (ERI-2000-01-11b). However, with three different sites running eMatrix in large development

---

26. RUP: Rational Unified Process
projects, and in particular the UMTS project, the need for a central point of control became more urgent.

6.6.1 The reference system

In January 2000 a new unit, SSES\textsuperscript{27}, was established at Ericsson. In SSES a number of SPMs\textsuperscript{28} were responsible for a method & tools area each. One of the responsibilities for SSES was to define and manage a reference system which would be the coordination point for the different domains. A principle called ‘coordinated flexibility’ for the reference system was outlined already in 1998 (LTX-1998-10-12). This principle was based on two assumptions:

- The autonomy of local activity domains to evolve the domain to their needs must be maintained.
- The coordination items needed to coordinate the different domains must be mandatory for all domains.

Thus, a federated architecture of information systems was proposed (Sage, 2001). The idea was that all mandatory and common parts of all the local domains would be implemented in the reference system [stabilizing core]. This system, which was called MARS (MAtrix Reference System), would evolve at a far lower pace than the local domains. New organizations would be able to use the reference system as a starting point for their own, local adaptations. No sharp project would be allowed to run in the reference system in order not to bias it towards the preferences of a particular project.

However, the federated architecture was soon questioned by influential actors in SSES in favour of a single, centralized domain in which all projects would run. This would mean that the three existing domains would be merged into one:

“We think that the effort for reconcile and coordinate the existing applications is not that high. It will be possible to have several customers in the same environment. We see an environment or a database more as a building which has different lodgers. Some services they will use together some services are individual and we see it more as a challenge to find a solution for this.” (MT5, e-mail, 2001-05-01)

A detailed business case was also made which showed that a centralized solution was the cheapest, albeit with some reservations:

“The most economical solution is the centralized eMatrix solution with a common shared business model and one database. This is also the most common alternative for eMatrix implementations in other companies. By using one centralized solution a company can harmonize its functions and create a common way of handling information. Different sites have in practise different requirements resulting that not all sites want or can share one common Business Model without adaptations. In this case a realistic solution is a combination of central and loosely coupled scenarios.” (ERI-2001-04-20, p. 20)

\textsuperscript{27} SSES: System and Software Engineering Support

\textsuperscript{28} SPM: Strategic Product Manager.
THE HISTORY OF THE FRAMEWORK

However, no effort was made to estimate the cost of achieving a shared meaning about the common business model (the Conceptual Model) and its implementation in the eMatrix system \([\text{shared meaning}]\). I claimed that this cost would reverse the conclusions in favour of a federated architecture, although I had no quantitative grounds for that claim. This discussion about a central architecture versus a federated one became a recurrent theme in SSES until the central solution finally took precedence over the federated one in 2002.

SSES also started a number of other activities. One such activity was to define a common Conceptual Model for Ericsson. The starting point was the Conceptual Model which had evolved at the S-domain. The first meeting was held in February 2000 which was followed by a series of meetings during the spring 2001. The premises for these discussions were that the model should be tool independent. This model, which was called SCIM (SSES Common Information Model), evolved only through lengthy discussions among the participants and was never tried out in practice. Eventually SCIM was productified and managed as an individual artefact. It was also suggested that the steering of SCIM would be done by the SPM’s:

“\text{I discussed how \{SCIM\} should be managed and controlled within SSES with N. His proposal was the SCIM product should be controlled by all SPM's in SSES.}” (E-mail, 2001-05-28)

This meant the Conceptual Model was detached from its implementation in eMatrix and that the control of the evolution of the model was moved from the actors in the coordination domains to well recognized managers who nevertheless had never worked with the model or eMatrix. The experiential learning aspect was never understood and thus largely ignored \([\text{experiential learning}]\). This trend was to become more proliferated during 2002.

Other activities concerned interfacing eMatrix to other information systems. A project was started in March 2000 to integrate eMatrix with ClearCase through a third party commercial product. However, due to performance problems this track was closed down in November 2001 and a simplified solution was implemented.

6.6.2 The Core Network starts using eMatrix

In January 2001 I started to work at the unit in Stockholm which was responsible for the Core Network part in UMTS. The core network consists of a number of nodes which perform various task in the network. My responsibility was to introduce eMatrix on the core network level. The strategy was to implement the requirements from the network level in the A-domain. In order to prioritize the requirements on that domain a Change Control Board (CCB) for eMatrix was initiated with me as a chairman. The first meeting was held in March, 2001. All through the summer 2001 discussions continued about what requirements to implement.

Although all parties had the ambition to compromise, it turned out that the requirements from the network level were quite disparate from the requirements on the nodes in the network. After five months virtually none of the requirements from the network level had been implemented. As a consequence the UMTS network level decided to move to the MARS C-domain. The CCB was closed on September the 10th and the responsibility for the A-domain was taken over by the Aachen unit. This experience was yet another indication that the effort
to agree on what should be implemented may become prohibitive if the activity domains are too disparate \[\textit{shared meaning}\].

6.6.3 Corporate on stage

During 2000 and 2001 the contacts with Corporate intensified. A large investigation about the future PDM solutions at Ericsson was carried out during the autumn 2000 where the eMatrix experience was one input. I presented the Framework at an all Ericsson meeting in Vienna in January. The future of eMatrix was discussed at a large meeting in March 2001.

In April 2001 a project was started to evaluate eMatrix and the C-PDM system. This was the first time these systems were officially recognized as ‘equals’ from Corporate. However, no decisive recommendation resulted from this investigation.

In late 2001 the budget responsibility for eMatrix was transferred to Corporate IT which meant that Corporate representatives were participating in the steering group of MARS for the first time. This meant that Corporate IT was now in control of the further evolution of eMatrix at Ericsson.

Discussions with Matrix-One Inc. concerning licenses had been initiated early 2000. On the 1st of May, 2001, a corporate agreement was signed and in December 2001 a joint press release from Matrix-One and Ericsson concerning this agreement was issued. This was still another cornerstone in the establishment of eMatrix at Ericsson.

6.7 From federalism to monarchy (2002)

During 2001 and 2002 the telecom crisis hit Ericsson hard. The staff was reduced drastically from 107,000 to less than 60,000 in a couple of years. The organization at the Core Network unit was shut down. I moved back to UAB in November 2001 where a new unit for methods & tools was established with the main purpose of reducing costs.

In February 2002 the radio network development unit at Ericsson started to use MARS. In March 2002 the S-domain was moved to the central MARS domain. The A-domain in Aachen was put on hold and the L-domain did plan to move to MARS.

During 2002 the trends from the previous year accelerated. In the new method & tools unit the separation between the Common Information Model (the Conceptual Model), the Work Package methods (the Process Model) and the eMatrix tool (the Information System) was institutionalized. For each one of these areas a technical manager was appointed with full control of their further evolution.

The work to define a common information model was organized as a regular project with four different tracks, a corporate steering group, a reference group consisting of 10 members from 8 different organizations and a coordination group consisting of 13 members from 7 different organizations (ERI-2002-07-31).

Thus, instead of continuing with the ‘coordinated flexibility’ strategy that had been proven successful at the three local domains, the course was steadily set for a ‘one-for-all’
centralized domain. In the words of Davenport et al. (1992), the path went from ‘federalism’ to ‘monarchy’:

“[Federalism] has a number of desirable features, and in today’s business environment, it is the preferred model in most circumstances. Its distinguishing feature is the use of negotiation to bring potentially competing and noncooperating parties together. [...] Each realm contracts with [...] other realms to cede some of its information assets in return for helping to create a greater whole. (Davenport et al., 1992, pp. 58-59)

“[Monarchy is when the] CEO, or someone empowered by the chief executive, dictates the rules for how information will be managed. Power is centralized, and departments and divisions have substantially less autonomy regarding information policies. [...]. As a model for information management, this means that domination is established over what information is collected, in what form, by whom, and for what ends.” (ibid., p. 58).

For my part, the story of the Framework evolution at Ericsson ended in June 2002 when I was dismissed from the company. In summary, the achievements were substantial. Between 1999 and 2002 approximately 140 main projects and sub-projects at Ericsson have used various parts of the eMatrix (ERI-2002-06-06). These projects were distributed over more than 20 different development units around the world and were carried out in a fiercely turbulent environment.

From my first contact with the lonely salesman in 1996 to a corporate strategic tool impacting major development projects at Ericsson in 2002 less than six years had passed.

6.8 Analysis of the Framework trajectory

In this section I will analyze the evolution of the Framework within Ericsson as described in the previous sections. The method I will use is inspired by the Actor Network Theory (ANT) (Latour, 1998). In ANT the social and technological aspects of human activity are seen as a unity, which is well in line with the praxis perspective in this dissertation. Humans and artefacts are both considered actors. For example, an information system may have interests of its own like any human actor. In order to underline this somewhat provocative position, the term actands are sometimes used instead of actors.

In ANT an innovation is considered to follow a certain trajectory in a society. This trajectory is called a program. The innovator inscribes certain intentions in the innovation. The inscription will change as the innovation becomes engaged in a heterogeneous actor-network of people, organizations, standards and artefacts. Eventually the innovation and its inscription may solidify into irreversible network elements called black boxes. The irreversibility means that the innovation has become an institution in the society and that it is no longer possible to go back to an earlier stage.

29. This means that my empirical data concerns mainly the first half of 2002 but to some extent also the last half of 2002 since I officially left the company on December 31, 2002.
30. For a discussion about ANT and information systems research, see Walsham (1997).
CHAPTER 6

The program is opposed by an *anti-program* which represents opposing interest groups and existing black boxes which might clash with the program. For example, a legacy information system may be part of an anti-program opposing the introduction of a new information system. In order to engage allies which may adopt the innovation, a series of *translations* of the innovation must take place. The purpose of a translation is to align the interests of other actants with the interest of the innovation. The inscribed pattern may not survive in the translation process since the anti-program may inscribe other intentions in the innovation. For example, the innovation may be used in ways not intended from the outset.

The trajectory can be illustrated in a diagram (as in Figure 43). The horizontal dimension, called *syntagm* in ANT, corresponds to actors which have adopted the innovation. The vertical dimension, called *paradigm* in ANT, corresponds to the translation that has taken place. An expansion in the syntagm dimension to the right must always be paid for by going down in the paradigm dimension.

Although the apparatus of ANT may seem awkward, it provides a suitable vocabulary to analyze the Framework trajectory. The Framework may be regarded as a program with this author as the innovator. The intention inscribed in the Framework was to pursue an improvement in the Ericsson coordination practice which is rooted in the dialectical worldview (see Taxén, 1995). The ultimate consequence was meant to be a set of irreversible black boxes with these intentions inscribed. How it in fact turned out is shown in Figure 43.

In the following sections I will go through each of the phases.

*A pattern emerges (1990-1995)*

In this phase, the ideas behind the Framework were articulated. The Information Flow Diagrams (the origin of the Process Model) and the Specification Based Data Model (the origin of the Transition Model) appeared. Also, in the work with process architectures the process
core was an early example of the Stabilizing Core (Taxén, 1995). At this point I was the only actand associated with the Framework. An anti-program could hardly be noticed.

*Incremental development (1996-1997)*

In the following phase the incremental development method package (IDMP) was developed and tried out in the CMS 30 Phase 7 project. This meant that more actands were involved in the program. The IDMP relied heavily on the Conceptual Model and also to some extent the Process Model. The experiences indicated that a better tool support was needed in order to manage the incremental way of working. By accident, I came across the eMatrix PDM system at a fair and eventually this system became the basis for the Information System. Together with the Domain Construction Strategy and the Conceptual Model, the Information System was used in a number of pilot projects with good results. My original intentions remained inscribed in the Framework and became during this period more elaborated and articulated.

However, the anti-program began to influence the trajectory at this stage. The main actands were:

- The C-PDM system: This was the only allowed PDM system at Ericsson, and since eMatrix was regarded as a PDM system, the Framework Information System was against Corporate policies. However, in the C-PDM system it was more awkward to change the implementation of the models which meant that the Domain Construction Strategy could not have been upheld.
- The organizational unit where I worked: This unit did not want a separate solution outside the Corporate strategy.
- The Rational initiative at Ericsson: Rational became responsible for the software development at Ericsson in December 1997, which resulted in dire consequences for the Framework program.

*Termination and despair (1998)*

In the next phase the program was on the brink of extinction. Although the first set of licenses for the Information System was bought, the budget for supporting the pilots was gradually squeezed. This meant that the pilots were on their own and eventually these programs ceased. The Methods & Tools unit in my home organization was closed down. For a period I kept the program going by giving demonstrations and trying to get funding from wherever. One sponsor was actually an insightful person in the Corporate IT unit (IT2). He believed that the incremental way of working was too valuable to get lost. This was an indirect criticism of Rational who claimed that the IDMP was not necessary. However, he wanted the C-PDM system to replace the eMatrix as the Framework Information System.

By and large, the anti-program had at this time more or less overhauled the program. At this point I was about to give up and my recollection of the first quarter of 1998 can be summarized in two words: utmost despair. However, more or less by accident the project manager

---

31. I was intrigued by a well-known Swedish stand-up comedian who fiercely stated that “It is never too late to give up”.
CHAPTER 6

for a new project named Beamon became interested (PM2). This was the turning point for the program.

Renewal in conflicts (1998-1999)
During the summer 1998 a prototype for configuration and requirement management was developed on the initiative of the Beamon project manager. The experiences from this work gradually matured into a conclusion to use the Information System in the Beamon project. The anti-program most outspokenly was the Rational one. The peak of the clash between the program and the anti-program was the meeting in November 1998. After some turmoil a decision was taken to use the Information System as a ‘gap-filler’ until Rational could provide an equivalent system, something that however never happened. In May 1999 the Beamon project started to use eMatrix and the first domain, the S-domain, at Ericsson was a fact.

Soon after three more projects joined the program. Besides the Information System, the Conceptual Model and the Domain Construction Strategy were used. Again, the original intention of the Framework remained inscribed and became more elaborated and articulated during this period. However, the severe stability and performance problems that were experienced with eMatrix made it temporarily part of the anti-program. If these problems had not been fixed, the program might well have ended here.

Other practices catch on (2000)
In the following phase two more domains were established, the A-domain in Aachen and the L-domain in Linköping. Especially in Aachen the Framework was fully utilized in the exceptionally complex development of the Mobile Switching Centre node in the UMTS network. The Process Model was a variant of the one in the Incremental Development Method Package called Work Package based development. The Transition Model, although never articulated, was apparent in the work to interface the Information System to a number of other information systems needed. The Domain Construction Strategy was heavily employed in the construction of the coordination activity domain.

In this phase the anti-program was suppressed since the Framework was backed by the project managers in very important and prestigious projects. The original inscription in the Framework was at this point at its peak. It must however be remembered that the three activity domains were only bridge-heads, albeit very important ones, in the overall Ericsson practice.

A central concern (2001)
A corporate agreement with the vendor of the eMatrix system was signed in May 2001. This turned the eMatrix system into an irreversible black box in the Ericsson practice. Also, during 2001 the need for a coordination between the three domains became apparent. As expected the Conceptual Model differed markedly between the domains. This created unforeseen problems since some projects ran across several of these domains. Discussions were started to set up a federated structure of cooperating activity domains. The idea was to establish a central reference system which implemented a stabilizing core of mandatory elements in the Framework. A product management function was set up to manage the reference sys-
THE HISTORY OF THE FRAMEWORK

tem and work was initiated to interface the reference system to other systems like software configuration systems.

The reference system, which was called MARS (MAtrix Reference System), turned however soon into another domain for ongoing development projects, the C-domain. This was not the intention with the reference system. The original inscription gradually changed into an inscription of being the one system backed by Corporate. Workshops were held to decide on a common Conceptual Model and the numbers of actors around MARS increased significantly.

By this translation the original program was overhauled by anti-program. The inscription of a federated structure of cooperating activity domains was replaced by the inscription of one, central system which every activity domain should use. The most apparent consequence was that the Domain Construction Strategy was replaced by a traditional, linear development strategy which slowed down the progress and increased the costs (see Chapter 7 Framework consequences). The close interaction between developers and users was broken and replaced with central reference and steering groups. One actor expressed this in the following way:

“... if you look at the functionality which has come out of the system it is incredibly little functionality during a very long time compared to the functionality we produced with two persons in half a year” (VDR1-102:44).

My interpretation is that the Framework never reached the state of an irreversible black box with the original inscriptions. Instead the Conceptual Model, the Information System and the Process Model began to appear as actants of their own. An obvious indication of this was that the Conceptual Model became an isolated product which was developed and maintained by discussion groups rather than by demonstrating its usefulness in practical settings.

From federalism to monarchy (2002)

In the last phase the development towards one central domain and central control accelerated. The upheaval in the telecommunication business during 2002 lead to a fierce cost reduction hunt. A central Ericsson wide control of the Information System, the Conceptual Model and the Process Model was established. Projects were transferred to the MARS system, the Stockholm activity domain was closed, the Linköping domain planned to move to MARS and the Aachen activity domain was put on hold. However, it remains an open question whether these measures will in fact save any costs.

The eMatrix system is now (2002) firmly established as a black box in the organization and used by a large number of projects and actors. The inscription associated with it has however turned into ‘a requirement management system for large projects’. This is a significant retreat from the original intentions with the Framework where requirements were but one type of coordination items among others.

The Conceptual Model and Process Model evolution have split in two branches. One branch is the one associated with the eMatrix implementation. These models continue to evolve with whatever is implemented in the eMatrix system. In the other branch these models have become products of their own which are developed and maintained as separate artefacts. As such, the models tend to solidify quickly into relics without any practical implications.
However, as signs for achieving shared meaning about the coordination context they may still play a role. It must also be kept in mind that the productified models, and in particular the Conceptual Model, would not exist without their forerunners during 1998-2000. This is a fact which is easily forgotten once the models have been institutionalized. There is simply no way a group of actors could have come up with these models by mere reflection. Any model is as good as any as long as its usefulness is not tested in practice.

However, the cardinal interaction between the Conceptual Model, the Process Model and the Information System in practical settings is broken. The actands in the original program, including myself, are more or less gone. The consequence is that the Framework program is now virtually extinguished at Ericsson and the original inscription is hardly discernible any more. Whether that can be revived in the future remains to be seen.
7 Framework consequences
In this section we describe and analyze the consequences from the intervention of the Framework in the Ericsson development practice.

7.1 Penetration into the Ericsson development practice
In this section we will describe the penetration of the Framework into the Ericsson development practice. A project is considered as impacted by the Framework if the project uses or provides coordination information via the Framework, mainly the eMatrix Information System. The delimitation of what constitutes a project is somewhat intricate since the projects are constantly redefined with respect to content, delivery plans and location. As rough indicators of an identifiable project we have used
- main projects releases,
- sub-projects to main projects,
- the product, the development of which is to be coordinated.

The number of impacted projects and sites are estimated around mid 2002 and documented in (ERI-2002-06-06).

Four different coordination domains were established at Ericsson in the following order: the S-domain in Stockholm, Sweden, the A-domain in Aachen, Germany, the L-domain in Linköping, Sweden and the central C-domain in Stockholm, Sweden (see Chapter 6 The history of the Framework). The actor involvement in each domain varied considerably. For example, a designer may be only occasionally impacted, while an actor working with the eMatrix application\(^1\) may be involved on a daily basis.

In summary, approximately 140 projects distributed over more than 20 sites worldwide have been impacted by the four coordination domains. Moreover, the time frame for this course of events has been surprisingly short. The first project, Beamon, started to use the Framework in May 1999; that is, the entire expansion until mid 2002 took place in less than 3 years.

7.2 The structure of coordination domains
In this section we will describe the major coordination situations in the domains and the different roles involved.

7.2.1 Coordination situations
As described, the intervention of the Framework resulted in four different coordination domains. In Figure 44 the Conceptual Model in the S-domain is shown (early 2000). A number of coordination situations are indicated (the ovals). Each of these situations can be regarded as a coordination domain in itself with a Conceptual Model, Process Model, Stabi-

\(^1\) By ‘application’ we mean domain-specific additions to the commercial eMatrix platform.
lizing Core, etc. As such, these can be constructed relatively independently of other coordination situations. For example, the situations which concern requirement management (the upper left oval) can be constructed independently of the Change Proposal situations (the lower left oval) as long as the relation between them is not changed (ChangeControl). This makes it possible for different situations to evolve at their own pace.

The coordination situations for the four domains have varied considerably and the constructed coordination domains are quite disparate with respect to all Framework elements. Some of these situations such as requirement management appear in all four coordination domains, albeit in different forms. Other situations may appear in some domains only such as incremental development in the form of work packages development. This was found only in the A-domain.

**Incremental Development**

Incremental development as conceived at Ericsson replaced the traditional waterfall development model. The linear, phased oriented way of working inherent in the waterfall model is split up into a number of increments which each provides an executable and testable functionality (see Figure 45). Internally each increment may employ the waterfall model or any other development model such as the Rational Unified Process (RUP). The main reason for the
transition to an incremental development is the need to manage unstable requirement situations, capture errors early and provide earlier feedback to customers and designers.

As described in Chapter 6 The history of the Framework, an Incremental Development Method Package was introduced in the Ericsson practice during 1996-1997 (Karlsson & Taxén, 1997). This package fell into oblivion during 1998. However, the essential ideas in the package were revived in the A-domain during 1999-2000. The main difference was that the term ‘feature increment’ was replaced by ‘work package’ and that the functional anatomy was subdued (see Section 7.3.3 Orientation). The work packages were also substantially smaller and developed in a shorter time than the feature increments.

Requirement management
This coordination situation concerns the management of requirements to be fulfilled by a certain product being developed. The requirements are either managed as documents called ‘requirement specification’ or as individual requirements. The latter is usually preferred since it enables complete traceability from individual requirements across to test results and deliveries of the product.

Status accounting
Status Accounting is the process whereby records are maintained of the current and historical state of the product being developed. The frozen items are collected in ‘baselines’ which are the manifestation of the achieved status of the product and its related items. Any change to an item in a baseline must be meticulously analyzed before being implemented.

Engineering Change Management
This coordination situation is about engineering change management, which is a process to manage requests for changes of items in a baseline. At Ericsson these requests are called ‘Change Requests (CR)’. The proposed change must be approved by a certified board after an investigation of the consequences.

Milestone planning and monitoring
This coordination situation concerns the planning and monitoring of the project. Each milestone defines the required status of a set of items which shall be reached at a certain date. A milestone is similar in structure to a baseline. The difference is that a milestone has a time dimension and that milestones are not involved in change management. This means that a certain item may be included in both a baseline and a milestone but for various purposes.

Test management
Test management concerns the management of the testing of the product. This is usually done by executing a number of test cases and recording the result of the tests. The test cases are designed in such a way that the requirements issued towards a product can be tested for fulfilment or not.

Product management
This coordination situation concerns the management of the product and its associated describing items, usually documents of various types. Product management is the traditional
CHAPTER 7

application area of Product Data Management (PDM) systems of which eMatrix is one. A typical function of these systems is to manage different configurations of the product.

7.2.2 Actor roles and groups

One of the elements in the Framework is the Stabilizing Core. This holds rules, norms, mores, etc., which provide stability to the coordination domain. As an example of such rules, the access rights for roles and groups in the S-domain are given in Table 15:

Table 15. Roles, groups and access rights in the S-domain

<table>
<thead>
<tr>
<th>Role - group - user / Access</th>
<th>Matrix System Admin.</th>
<th>Change Business Model</th>
<th>Define new users</th>
<th>Delete objects</th>
<th>Manipulate objects</th>
<th>Create CR, CR Comment</th>
<th>View all</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project System Administrator (role)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Project Business Administrator (role)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Project User Administrator (role)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Project Full User (role)</td>
<td>Requirement Coordinator (role)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Configuration Manager (role)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>CR Receiver (role)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Project Manager (role)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Functional Tester (role)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>System Manager (role)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Team Leader (role)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Beamon (group)</td>
<td>Change Control Board (group)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Change Request Analysis (group)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Project Managers (group)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Hardware design UAB (Stockholm)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Hardware design TEI (Italy)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Hardware design EPA (Australia)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Software design UAB (Stockholm)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Software design TEI (Italy)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Software design EPA (Australia)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Matrix vendor</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>non-Beamon (user)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>guest (user)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

2. Includes user “creator”
3. Access rights are assigned to role “Project Full User”
4. Restricted to objects of type “REQUIREMENT”
5. Restricted to objects of type “Baseline Definition”
6. Restricted to objects of type “Change Request” in state “PENDING”
7. Restricted to objects of type “MILESTONES”
8. All project members belong to this group. Access rights are assigned to group “Beamon”
9. Anyone that may issue a CR outside Beamon

Like the other elements in the Framework, these access rights are experientially determined. The key issue here is maintaining a balance between too tight control and too loose control.

158
Furthermore, as can be seen, the access rights are depending on the state of the object. For each state about 30 different types of rights may be assigned in eMatrix. Since there may be many types and many states for each type, the number of possibilities of assigning rights quickly becomes very large. This makes the definition of access rights one of the most cumbersome tasks in the construction of the coordination domain.

7.3 Consequences

In this section we will describe the consequences of the Framework intervention. The analysis method is described in Section 2.2.7 Research processes, strategies and methods, p. 43. A number of effect categories are defined. Each effect category is grounded in one or several empirical findings together with our interpretation of why a particular datum is associated with a category. For example, in Figure 46 the effect categories associated with the quotation PM4-05-3:33 is shown:

<table>
<thead>
<tr>
<th>Effect category</th>
<th>Description</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>PM4-05-3:33</td>
<td></td>
<td>The size of the coordination task enhances the use of a tool. Similar experiences are in CMS 30/97.</td>
</tr>
<tr>
<td>Effect category</td>
<td></td>
<td>Matrix is needed to support the work package method.</td>
</tr>
<tr>
<td>Effect category</td>
<td></td>
<td>Argumentation: we need the I.P. concept to succeed with the 15 projects, but the way of working is not possible without a proper tool support. The tool support has provided that tool support in several projects. Thus, a necessity for Ericsson to get the 15 projects out.</td>
</tr>
<tr>
<td>Effect category</td>
<td></td>
<td>There are a lot of dependencies. Without tool support (A) there cannot be handled (C). The dependencies must be managed in a tool.</td>
</tr>
</tbody>
</table>

Figure 46. Grounding of effect categories

This means that each effect category will eventually be grounded in several empirical findings. For example, the effect category ‘dependencies’ is grounded in 15 empirical datum out of a total of more than 600. Thus the effect analysis is grounded in a rich empirical material. Altogether, about 120 effect categories has been identified. The grounding of each effect category is given in Appendix 1. This step in the analysis corresponds to the open coding step in the Grounded Theory (GT) (Strauss & Corbin, 1998).

The consequences are derived from the impact of the Framework on a set of consequence categories. Each consequence category is associated with a number of effect categories. Our interpretation of the consequences is also given, that is, why we have associated an effect category with a certain consequence category. The Prerequisites - Action - Consequence schema in the GT approach is used, where the Prerequisite is interpreted as the Ericsson development practice, the Action as the Framework intervention in that practice and the Consequence the consequences from that intervention. This step in the analysis corresponds to the axial coding step in GT.

It should be noted that an effect category can be associated with different consequence categories, which means that the interpretation differs depending on the consequence category.
Thus, a consequence category shall be apprehended as a focal category related to a number of effect categories. The selective coding step in GT is not explicitly used. However, in Chapter 9 Discussion of the Framework approach, the results are discussed on a more general level.

In the following, each consequence category and its related effect categories are given in a tabular form. The column ‘#’ in the following indicates the number of grounding data for each category. For each category we also discuss some of the consequences and give examples of its grounding data. The complete grounding information and our interpretations are available in the research database, which means that it is possible to trace any interpretation to its grounding data. The tables are generated from the research database.

### 7.3.1 Sine qua non

The consequence category ‘sine qua non’ refers to the necessity of the Framework for particularly complex development projects. It is associated with the following effect category:

<table>
<thead>
<tr>
<th>Effect category, description</th>
<th>consequence</th>
<th>#</th>
</tr>
</thead>
<tbody>
<tr>
<td>sine qua non, Necessary pre-requisites</td>
<td>The Framework was necessary for the development of the MSC node in the 3rd generation mobile systems.</td>
<td>4</td>
</tr>
</tbody>
</table>

‘Sine qua non’ can be translated as ‘indispensable and essential conditions’. The most notable consequence from the Framework intervention is that experienced project managers claim that the development of certain nodes in the 3rd generation of mobile systems would not have been possible without the work package method and its implementation in eMatrix:

*R: Yeah, especially for the execution part I think we would not have been able to run this project without the tool. I think if you simply look at the number of work packages, the number of products that we have delivered, the number of deliveries that we have had, if we would have to maintain that manually, that would have been a sheer disaster. [...] we had some, only in my part of the project, some 200 work packages or work packages groups or whatever you want to call them, deliveries, on the average 2-5 subprojects within them 5-10 blocks being delivered, just keeping track of that [...] would have been a hell of a job." (PM4-05-1:07).*

---

2. Implemented in eMatrix.
FRAMEWORK CONSEQUENCES

7.3.2 Evolution

The consequence category ‘evolution’ refers to consequences regarding the continuous evolution of the coordination domains. It is associated with the following categories:

<table>
<thead>
<tr>
<th>Effect category</th>
<th>description</th>
<th>consequence</th>
<th>#</th>
</tr>
</thead>
<tbody>
<tr>
<td>agile teams,</td>
<td>Issues related to small and independent teams.</td>
<td>The agile teams in the S-domain made a fast evolution of functionality possible</td>
<td>1</td>
</tr>
<tr>
<td>change</td>
<td>Issues related to change in the environment</td>
<td>Re-planning due to changes are easier to do.</td>
<td>2</td>
</tr>
<tr>
<td>daily tuning,</td>
<td>Tuning of the eMatrix implementation</td>
<td>Small changes can be done on the fly during production use. Major changes should be avoided.</td>
<td>3</td>
</tr>
<tr>
<td>evolution,</td>
<td>Evolution: gradual changes in the AD</td>
<td>A coordination domain can be gradually evolving over time. This includes the Information System support.</td>
<td>30</td>
</tr>
<tr>
<td>feedback,</td>
<td>Issues related to feedback on events</td>
<td>Instant feedback from users which speeds up the implementation. Lack of instant feedback result in longer development times. Faster feedback on writing CR's</td>
<td>14</td>
</tr>
<tr>
<td>IDMP,</td>
<td>Incremental development according to Incremental Development Method Package.</td>
<td>Step-wise development according to IDMP is supported.</td>
<td>3</td>
</tr>
<tr>
<td>IS - eMatrix,</td>
<td>The eMatrix product data management (PDM) system which was used in the Framework</td>
<td>The ease of making changes in eMatrix makes it possible to continuously evolve the construction of the coordination domain.</td>
<td>30</td>
</tr>
<tr>
<td>transcendence,</td>
<td>The dialectics between tradition and renewal</td>
<td>The transition to a new way of working was hard on many occasions, for example, the change from managing requirements in a document to manage each individual requirement in the eMatrix tool.</td>
<td>4</td>
</tr>
<tr>
<td>work package</td>
<td>Issues related to the work package method</td>
<td>The work package method makes it easier to re-plan the development project.</td>
<td>9</td>
</tr>
</tbody>
</table>

New knowledge acquisition should be done gradually and build on an existing knowledge base. In all coordination domains the intervention of the Framework was done from an existing knowledge base. One reason for this was the experiences from previous, unsuccessful attempts to introduce new methods and tools. For example, in the S-domain the introduce of the Quality Function Deployment (QFD) method failed:

“But I guess that it is a combination of things which caused the failure. One is our own competence to use OFD, there was really no one here that knew the method in a good way. Because of that we might have had problems to transform and decompose the requirements in these different QFD houses.” (PM2-06:36).

From this experience as well as others, a major requirement was put on the Framework in the S-domain: it should be possible to adhere to known ways of working. One example of this is that the Change Request process evolved from the existing, document oriented way of working:
“RM: Because that was the point with eMatrix, that we should be able to take our process, which really wasn’t that well documented at that time, and adapt it to the tool. And the tool should be able to do the job that we... that if we wanted it to work in this way, then it should be possible to adapt the tool to that.” (CM1_2-06:56)

Similar examples can be found in the other domains which show that it is quite feasible to base the Framework intervention on an existing knowledge base.

*Changing customer requirements*

The intervention of the Framework had consequences on the ability to re-plan projects in response to changing requirements, for example from the customer. The ever increasing turbulence of the market in general and the 3G market in particular made re-planning a major issue in the projects:

“I: Did you also suffer from like changes in the project? R: We have a record from R8 who had to deal with 400 CRs [Change Requests] on functionality, but they still I think have to cope with 80 CRs, 80 to a 100 CRs on functionality and some of them are smaller changes but every change is impacting this baseline functionality. I: need some kind of re-planning and... R: yes. And then you need to have a quick process of re-planning, stabilizing again and communicating a clear picture that this is what is going to be on the build. This is the build content. Everything is in the AXE world driven around the build content from a technical coordination test perspective, preparation work. This is the key for the project, this is the heart beat, the builds are the heart beats of the project. I: So in some sense the tool and the procedure have been valuable? R: Yes, I think so, very valuable. But it is a long way to go.” (PM3-41:40b)

The incremental way of working in the form of the work package method and eMatrix was a cornerstone in building up the capacity for re-planning. Without these the complexity could not have been managed:

“I: But what you are basically saying is that all these dependencies in the way of working with the work package concept really needs a tool to keep track of... R: Yeah, yeah... if you just do the multiplication we had some 200 work packages, an average 2-5 subprojects and an average 5 blocks amounts to an enormous number...” (PM4-05-3:33)

*Daily builds*

Two types of continuous evolution of the coordination domain can be identified. The first one has the character of ‘daily build’ where changes are made more or less on the fly during an ongoing project. This is the dominant type during the exploration phase of the Domain Constructions Strategy (see Section 7.3.6). The reasons for the daily build type of evolution are mainly the achievement of shared understanding, fixing of errors and adapting to new needs during the project:

“I(35:54) Did you have any idea how many changes you have made during these years, since you first started to use the tool? How many attributes have been changed or any other things, any kind of change? R: 75% of what we have today. I: and in terms of number of changes, do you have any kind of estimate? [...] R: Maybe we 20-25 object now, I haven’t checked it and the WP [Work Package] has about 20-30 attributes, yeah nearly 200 changes I think. That’s growing, the overall model has grown a lot so, this requirement traceability came in later, first we had only this WP...”
anatomy part and we had this TCM [Test Configuration Management] handling part where we have LSV’s [Latest System Verified]. Then we had this requirement traceability, then we had the connection towards MHS [a tool for managing trouble reports] and I think you have required the connection to TEPOS [a test management tool] so this model will grow.” (MT3-35:54)

A necessary prerequisite for the daily build type of changes is that the implementation in the Information System is extremely easy to change, which is the case with eMatrix. However, it is important that the applications built on the flexible eMatrix platform are not built in such a way that the daily build evolution is hampered.

The daily build has the drawback that the application may turn out to be a patch work which has to undergo major reconstructions:

I: This way of working, how would you evaluate this? R: I think it has good and bad sides. The good side certainly is that you quickly solve the issue, the bad side is that it takes a lot of rework once in a while and if you do this for too long a time you can have a patch work design in the end. This is what we were suffering from like after a year almost. That we had a lot of redesign done and they could not properly clean up the system because they always pushed for something else and they always had to firefight for the next problem. So in the end, I think in the early 2001, they came back with the message that we need to redesign to save the installation, otherwise we might be in the danger to loose it.” (PM3-17:45)

The experiences from the existing coordination domains are so far that these problems can be mastered.

Long-term evolution
The other type of evolution concerns a long-term evolution which is related to the consolidation of the application and external changes. This may include new coordination situations, possibly replacing immutable mobiles in the organization. One such example from Ericsson would be to replace the existing trouble report system. This is now based on a stand alone information system (or rather a set of information systems) which in principle would be straightforward to implement in the Information System in the Framework. The ground for such changes has been laid in the existing coordination domains:

“Well, the positive effects that we have, that is that we have an integrated project support system where we have tremendous possibilities to improve, continuously improve our operations.” (PM2-45:43)

The long term evolution concerns all the elements of the Framework, that is, the Conceptual Model, the Process Model, the Stabilizing Core, etc., as well as the interactions between these. By focusing on the coordination domain as a whole, it is possible for these elements to evolve in concordance with each other. However, if the unit of evolution is reduced to the individual elements in the Framework, say the Information Systems or the Conceptual Model, the control of the evolution will be lost. This is what happened during the later phases of the Framework intervention at Ericsson (see Section 6.8 Analysis of the Framework trajectory, p. 149).
CHAPTER 7

Transcendence

The contradiction between tradition and transcendence is well-known in the literature. For example, Ehn expressed this in the following way:

“[If] I should single out one aspect that to me seems to be the most crucial to design philosophy, it must be [...] the dialectics of tradition and transcendence in design and use.” (Ehn, 1988, p. 127)

The most salient expression for this dialectic in the Framework concerned the transition to use eMatrix instead of adhering to the traditional document based ways of working. Some of these transitions were profound, for example the management of individual requirements instead of requirement specification documents. This transition necessitated a reversal in the traditional thinking. The requirement specification document, which previously was the coordination item, now became a list which was generated from the new coordination items – the individual requirements.

In other cases the transition contradictions could be solved by hiding them from the user. For example, the introduction of the so called Lean Client for managing Change Request meant in essence that the user only had to change the ULR from the traditional tool to the eMatrix Lean Client. The appearance on the screen was more or less the same.

As seen from a semiotic point of view, the transition to eMatrix mostly concerns the signifier aspect of the sign. For example, in the traditional way of working the Change Request handling was signified by various documents with particular document class numbers as signifiers3. In eMatrix the signifier was basically an icon. The signifying aspect was however in both cases ‘Change Request’. Thus the transition implies in essence a signifying process in which actors gradually come to associate the icon with ‘Change Request’.

The transition to a new way of working was hard on many occasions. The ‘daily build’ evolution caused problems for the users:

R(07-2:05) One of the main problems was that the eMatrix model was under development all the time because we started with the small part, then we took this and we took that and it kind of grew. Sometimes it also meant that we had impacts on things from before so that had to be changed and I had to make sure that I updated the work instructions that I had written previously. I: Did that cause a lot of frustration or anger or resistance in using the tool or... R: I don’t want to say a lot but of course there were some people nagging about that.” (MT4-07-2:05)

One extreme example of user resistance was the product manager ‘A’. In the S-domain requirements were managed as individual items instead of the stating them in a traditional requirement specification. ‘A’ did not approve of this. The following mail-conversation took place (‘M’ is the project manager of the Hercules project):

3. A Change Request has the document class ‘109 20’ in the Ericsson internal classification system.
FRAMEWORK CONSEQUENCES

“Hi, Who have come up with the idea to use eMatrix? Are those persons called to the meeting? I expect a presentation from them on why and how eMatrix is intended to be used in Hercules in the requirement, design and verification phase. Best regards. /A (e-mail 1999-12-06)

“M. Would you please inform A and other new persons in Hercules about the decisions to use eMatrix, so we do not have to spend time discussing decisions already taken at the meeting on Thursday. Regards /Lars” (e-mail 1999-12-06)

“Hi, Well, if there are no answer to why and how there is no use for me to attend the meeting. /A” (e-mail 1999-12-06)

The result of this exchange of opinions was that ‘A’ refused to have anything to do with the eMatrix tool unless a requirement specification report could be generated which had exactly the same layout as the traditional report (45 mm left margin, 23 mm right margin, etc.). Fortunately, this kind of reaction was rare.

7.3.3 Orientation

The consequence category ‘orientation’ refers to the ability of actors to orient themselves in the coordination domain. It is associated with the following effect categories:

Table 18. Consequence category – Orientation

<table>
<thead>
<tr>
<th>Effect category</th>
<th>description</th>
<th>consequence</th>
<th>#</th>
</tr>
</thead>
<tbody>
<tr>
<td>anatomy,</td>
<td>Issues related to anatomies</td>
<td>The implementation of the anatomy in eMatrix enhances orientation.</td>
<td>7</td>
</tr>
<tr>
<td>dependencies,</td>
<td>All kinds of dependencies, closely related to traceability</td>
<td>All dependencies in the Conceptual Model can be visualized in eMatrix. This enhances the orientation.</td>
<td>15</td>
</tr>
<tr>
<td>focal change</td>
<td>Issues related to focal changes</td>
<td>Changing focus makes orientation easier.</td>
<td>3</td>
</tr>
<tr>
<td>orientation,</td>
<td>Issues related to orientation, see the whole picture, understand relations</td>
<td></td>
<td>9</td>
</tr>
<tr>
<td>overview,</td>
<td>Overview of a complex situation</td>
<td>Overview is vital to orientation. The implementation of the Conceptual Model in eMatrix provides better overview than traditional mechanisms.</td>
<td>4</td>
</tr>
<tr>
<td>traceability,</td>
<td>Issues related to traceability</td>
<td>Traceability is a kind of dependency</td>
<td>14</td>
</tr>
<tr>
<td>transparency,</td>
<td>Issues related to transparency, to see how things are related.</td>
<td>Various mechanisms in the eMatrix implementation, for example, reports enhances the transparency.</td>
<td>14</td>
</tr>
</tbody>
</table>
Anatomy:
The anatomy is a model of the functional dependencies in a system. It shows the order in which the functions must ‘come alive’ in order for the whole system to be operational. For example, in Figure 47 the ‘Load OS’ function must first be executed in order for the ‘HW initiation’ and ‘Load Appl’ functions to execute.

The anatomy is used extensively for planning and monitoring purposes in the system development practice at Ericsson. It is a key element in any variant of incremental development:

"And also based on the anatomy chart, which is also I think a very important mechanism, you see a lot of dependencies both time wise and product wise, etc. So you can fix things that need to be fixed first and fix problems that are at the end of the project so to say at a later stage. Of course that is also a key issue." (PM4-07-0:37)

An increment may be defined in terms of which functions in the anatomy will be developed in the increment. In the work package method the functional dependencies are subdued and only the dependencies between increments / work packages are managed. An example of a work package based anatomy is shown in Figure 2, p. 34.

The consequence of the Framework intervention is that it is possible to utilize the anatomy for coordination and controlling purposes. The anatomy is modelled by defining a recursive relation between functions or work packages in the Conceptual Model. This is directly implemented in eMatrix. The anatomy may be apprehended as a variant of the Specification Based Data Model (Gandhi & Robertson, 1992) where the implementation element is suppressed. Since the Transition Model in the Framework is based on the SBDM the anatomy is included in the Transition Model (see Section 9.1.4 The Transition Model, p. 218)

Dependencies
In a complex situation it is important to have control over the dependencies between coordination items. The Conceptual Model and its implementation in eMatrix make it possible to monitor all dependencies between coordination items in the Conceptual Model:

"R: I think for the MSC [Mobile Switching Centre node] there is a clear need for the tool and I have met now a couple of people who have said: without the tool we would not have survived the CN [Core Network] projects. The tool eMatrix and CC [ClearCase]. I: Is that due to the complexity of the node itself? R: yes, the complexity of the node, the complexity of the dependencies, time pressure people have. If you visualize dependencies it is far easier to take decisions." (PM3-41:40a)

4. It can be noted that the work package anatomy in Figure 2 is turned upside-down as compared to the functional anatomy in Figure 47. This is still another indication of the difficulties of achieving a shared understanding in the organization.
FRAMEWORK CONSEQUENCES

Focal changes
Orientation in a complex situation is made easier if the attention can be directed alternately to different foci while still keeping the whole complex together. The way the Framework Models are implemented in eMatrix makes it possible to direct the attention to any foci inherent in the Models. This is illustrated in the screen dump below from eMatrix (Figure 48) where the immediate surroundings of any function can be seen by changing the focus.

![Diagram of focal change]

Another example is that dependencies are traditionally expressed in documents as references to other documents. In eMatrix these dependencies are directly visible, either by navigating in eMatrix or by disclosing them in a report. This also improves the orientation:

“R: You can look at different levels because of... how shall I say that.... you can look at it on a very overall level and you go down to specific as you want to up to document status.” (MT4-14-2:47)

Traceability
Traceability is a particular kind of dependency where the purpose is to follow a path of dependencies from for example requirements to customer deliveries. Again, the Conceptual Model and its implementation in eMatrix make this possible:

“We have a good support for configuration management, our engineering change requests are in order, our baselines are in order, etc. In addition I think that we have the possibilities to manage requirements in a good way and make them obvious and we can achieve a very clear traceability all the way from customer requirements one might say.” (PM2-45:43)

Transparency
By transparency we mean the ability of actors in the coordination domain to apprehend relations between items. This can be interpreted as a signifying process in which the relations in the Conceptual Model are signified for actors which have not been directly involved in the construction of this model. Thus, in their practice they do not see the Conceptual Model directly, only the impacts of it in terms of other signs signifying relations, for example as reports from eMatrix:
CHAPTER 7

“R: Before every role maintained a piece of information it was responsible for. But in the end all pieces put together, they should build an overall picture and what eMatrix enables us to get, this full picture also to cross the border and see “aha this is information somebody else in another role thinks is connected to this one” that is a complete picture of the overall view and not just the limited view the person is responsible for. That is the main benefit I think. I: And in what ways does that happen? R: In the reports for example. By having another column showing the connected objects of another type.” (MT3-50:55)

An example of such a report is given in Figure 49:

![Figure 49. Report from eMatrix showing dependencies](image)

**Overview**

This category may be interpreted as the ability to get a clear overview of how things are related. The Framework is capable of producing overviews that are not possible to achieve with the traditional mechanisms in the Ericsson practice:

“I(16-2:05) How would you actually do this before you had the tool, the same kind of things? Was it possible at all to do this? R: No, actually it wasn’t. We had a simultaneous project running, it was called British Telephony Server […]. I also had a lot of contact to that project manager there. And he said that he was really lacking an overview. He said I find it very difficult now to find out with all the change requests coming in, what is actually the scope of a specific sub-project. And what it is exactly what I have in my scope. He really missed the overview in that way sometimes in the project. And that was also a large project.” (MT4-16-2:05)
7.3.4 Separation of concerns

The consequence category ‘separation of concerns’ refers to the consequences of the Framework intervention that can be associated with separation of concerns. It is associated with the following effect categories:

Table 19. Consequence category – Separation of concerns

<table>
<thead>
<tr>
<th>Effect category</th>
<th>Category description</th>
<th>Consequence</th>
<th>#</th>
</tr>
</thead>
<tbody>
<tr>
<td>anatomy</td>
<td>Issues related to anatomies</td>
<td>The anatomy provides a way to separate a complex problem into separate contexts</td>
<td>7</td>
</tr>
<tr>
<td>contextuality</td>
<td>The situated aspects of an activity domain</td>
<td></td>
<td>14</td>
</tr>
<tr>
<td>focus</td>
<td>Issues related to focusing efforts on one target</td>
<td>Easier to pin-point the source of problems</td>
<td>3</td>
</tr>
<tr>
<td>focal changes</td>
<td>Issues related to focal changes</td>
<td>Possible to switch between levels of abstraction</td>
<td>3</td>
</tr>
<tr>
<td>IS federations</td>
<td>Issues regarding one or several databases for eMatrix</td>
<td>No federated structure between the four eMatrix environments</td>
<td>16</td>
</tr>
<tr>
<td>isomorphism</td>
<td>Mappings between world views</td>
<td>Different backgrounds between users and developers caused problems.</td>
<td>6</td>
</tr>
<tr>
<td>one management tool</td>
<td>Issues related to one tool (eMatrix) to support coordination</td>
<td>The entire Conceptual Model can be implemented in one tool.</td>
<td>23</td>
</tr>
<tr>
<td>responsibility</td>
<td>The pin-pointing of responsibility to the proper place.</td>
<td>The identification of responsibilities is more evident</td>
<td>14</td>
</tr>
<tr>
<td>discrimination</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>separation of concerns</td>
<td>Categories related to hiding unimportant phenomena in a context</td>
<td></td>
<td>20</td>
</tr>
</tbody>
</table>

All these effect categories are in some way or another related to the separation of a complex situation into contexts which are meaningful and comprehensible for the actors. Here we will discuss some of the effect categories. For a discussion about the ‘one management tool’ and ‘IS Federations’ see Section 7.4 Information System Architectures).

Isomorphism

Isomorphism concerns how different world views may be consolidated. Unless this is done, problems may occur:

“I: Did you notice any major sort of conflicts or issues that could not really be solved? R: Maybe something that the people that are doing this implementation in eMatrix did not really have a project experience. So there were quite a lot of explanations needed… and trying to convince why some things were needed. I think that was one of the key issues. Of course people developing tools and processes in general they have a different background, they are not really experienced at project and that I think was one of the issues and caused a lot of discussions that we had. Why is it important that we want to have certain things in place.” (PM4-13-3:12)

As discussed in Chapter 8 Reinforcement rods, isomorphism needs a common frame of reference onto which different world views can be mapped. The intention with the Transition Model is that it should provide such a frame of reference. The most salient expression for this is the use of the work package method in combination with the anatomy. From the point of view of total project management there is no interest in the interior of each work package.
The knowledge interest lies in the status of each work package and how they are interrelated. On the other hand, from a single work package point of view, the focus is on the interior of the work package; on how to develop the required functionality in time. Thus, the work package specification is the bordering element between the total project management and the work package developers.

**Separation of concerns,**

The separation of concerns into manageable parts must be done in such a way that the parts still are joined in a meaningful whole. Moreover, in order to grasp both the parts and the whole, the possibility of changing foci between parts and whole is beneficial:

“Of course there is also something that is maybe also a benefit of the tool is that we do have different levels of projects. And the tool can really provide support for various levels… You have the UMTS project and the MSC [Mobile Switching Centre] node project which is another level, you have main projects taking care of certain parts of the system and then subprojects which take part of certain subsystems or AM [Application Module, a structuring concept in the AXE system]. And all those things are also recorded in one tool. Of course there are different requirements depending on the level of projects. That is a benefit as well, you have one common object model, one common database for all those levels which could get their input from and that is something very valuable…. R: It is one common database with everything in it. Its just a matter what kind of information you are interested in. You don’t have to summarize all the data from lower levels into reports for higher levels, its all in the tool and you can ask some mechanism to load it, that is very valuable.” (PM4-08-3:55)

**Distribution of responsibilities**

‘Distribution of responsibilities’ refers to the consequences of the Framework for identifying responsibilities. One consequence from the Framework intervention is that it has become more difficult for actors to escape their responsibility by pointing in other directions:

“R: There was always a huge debate whether there is a problem in the build content list or not. People usually said… if you talk to development projects they always said we don’t have a problem, it’s him. I will never forget I was in a meeting where they were actually pointing in a circle. So at the end of this meeting everybody said “It’s not my problem, it must be somebody else!”. So nobody in this meeting had a problem. Nobody could even think of having being even close to a problem. But at the end of the day the build process didn’t work. So obviously somebody must have had a problem. And they said no, they cannot believe it. And without a proof that it is their problem they would not do anything about it. So they were not even willingly to improve. Now this tool makes it so damn visible that they have a problem… ” (PM3-75:26a)

Another consequence is that the need for the total project management to engage in coordinating sub-projects has decreased:

“Yes, what is of course also the great benefit, and that is also the feedback we get from other subprojects, is that you have one common place where all the project area stored the information. It’s very easy to look up certain things like PA1 [Product Area] needs the status of a WP [work package] from PA2 and it’s simple to go to the tool and see what it is all connected to. It means that a lot of the coordination which previously went via the main project, now can go directly. It’s
FRAMEWORK CONSEQUENCES

closely tied to the work package concept of course. A lot of coordination is now happening on the level it should be and that’s on the subprojects level and not via the main project any more." (PM4-06-2:29a)

7.3.5 Centralism - decentralism

The consequence category ‘centralism - decentralism,’ refers to the contradiction between local or central control of activity domains. It is associated with the following effect categories:

Table 20. Consequence category – Centralism - decentralism,

<table>
<thead>
<tr>
<th>Effect category</th>
<th>description</th>
<th>consequence</th>
<th>#</th>
</tr>
</thead>
<tbody>
<tr>
<td>adaptation,</td>
<td>The adaptation of the tool, processes, models, etc.</td>
<td>Adaptation according to local needs is necessary. It is an open question whether that can be done in a central eMatrix database or if a federated network of eMatrix databases is needed.</td>
<td>5</td>
</tr>
<tr>
<td>balance,</td>
<td>The balance between central control and local autonomy</td>
<td>An example where a proper balance must be upheld is the assignments of access rights for users to perform operations in eMatrix.</td>
<td>3</td>
</tr>
<tr>
<td>centralism - decen-tralism,</td>
<td>Centralistic versus decentralistic tendencies in an organization</td>
<td>The first three coordination domains using the Framework were completely decentralized with no interaction whatsoever. Now (2002) the strategy is to go for one central coordination domain.</td>
<td>6</td>
</tr>
<tr>
<td>flexibility,</td>
<td>The ability to align the properties of an activity domain to needs for change.</td>
<td>Flexibility is a function of the stabilizing core. If the core is too encompassing or too minuscule flexibility is hampered.</td>
<td>9</td>
</tr>
<tr>
<td>IS federations,</td>
<td>Issues regarding one or several databases for eMatrix</td>
<td>Federated databases are one way to balance central and local control. The Framework makes it possible to implement a federated structure of coordination domains.</td>
<td>16</td>
</tr>
<tr>
<td>stabilizing core,</td>
<td>The stabilizing core of an activity domain</td>
<td>The stabilizing core is a placeholder for balancing mechanisms such as rules, norms, standards, etc.</td>
<td>4</td>
</tr>
</tbody>
</table>

The issue of local versus central control has been prevailing all throughout the Framework history. There is an inherent contradiction between the decentralized organization at Ericsson and an outspoken urge to use common processes, tools, rules, etc., throughout the organization:

“R: We can state that the decentralization of... the method and process work that we have today that our units within the Ericsson company that have their own subsidiaries work very differently, that we work very differently within one and the same company. Even if we should know this after this many years so, instead of consolidating into a unified way of working, it has diverged into 40 different ways of working.” (PM2-33:13a)

The Domain Construction Strategy in the Framework assumes that the construction of the coordination domain is done in the development practice in close interaction between the actors. This strategy was applied in the S, A and L domains but not in the C-domain. As indicated above (Section 7.1) this strategy was very successful. However, it had the draw-
back that these domains evolved as isolated islands into quite differently structured domains.

The intention with the reference system (Section 6.6.1 *The reference system*, p. 146) was to establish a stabilizing core for the different sites which would enable a balance between mandatory elements and elements under local control. Examples of mandatory elements are product identification rules, state chains for products, etc. The resulting architecture would have been a federated one in which several coordination domains cooperated.

The reference system was however never established in the intended way. During 2001 an investigation was made which showed that a common, central coordination domain was the cheapest one (ERI-2001-04-20). However, only tangible costs were included such as cost of licences, support, server hardware, etc. Other, more intangible costs such as the cost of achieving a shared meaning about the central domain were not included. Nor were costs related to evolution of the coordination domain.

The strategy now at Ericsson is steadily set for a common coordination domain, the C-domain. However, doubts remain if this is a viable way to go:

> “I know there are some initiatives and try to come up with one object model for all projects within Ericsson. And I think that’s a little bit trying to search for the Holy Graal. [...] Of course we are one company and of course we are having common processes. But trying to come up with one tool, one Holy Graal that covers everything, I think that is way too ambitious at this moment. I mean you can even see how different the ways of working are within our subprojects at this moment because of the characteristics of the products and the areas where they are located and so on. To even keep one common way of working for that is really quite a challenge. So if you try to come up with one way of working in the projects within Ericsson, a common object model for that, I think that’s way too ambitious at the moment.” (PM4-11-01:20a)

The consequence of this strategy is an open question.
FRAMEWORK CONSEQUENCES

7.3.6 Deployment

The consequence category ‘deployment’ refers to the consequences of the introduction, diffusion and absorption of the Framework in the Ericsson practice. It is associated with the following effect categories:

<table>
<thead>
<tr>
<th>Effect category</th>
<th>Description</th>
<th>Consequence</th>
<th>#</th>
</tr>
</thead>
<tbody>
<tr>
<td>agile teams,</td>
<td>Issues related to small, independent teams without reporting obligations to reference groups, steering groups, etc.</td>
<td>Agile teams contribute to a faster deployment.</td>
<td>1</td>
</tr>
<tr>
<td>champion,</td>
<td>Issues related to the champion role in the deployment</td>
<td>A champion in the project is a prerequisite for fast deployment.</td>
<td>2</td>
</tr>
<tr>
<td>deployment,</td>
<td>Issued related to deployment of the Framework in an activity domain.</td>
<td></td>
<td>43</td>
</tr>
<tr>
<td>establishment,</td>
<td>Issues related to the establishment phase in the Domain Construction Strategy.</td>
<td>An establishment of a coordination domain in one sharp project must be performed before the expansion phase.</td>
<td>6</td>
</tr>
<tr>
<td>expansion,</td>
<td>Issues related to the expansion phase in the Domain Construction Strategy</td>
<td>A full expansion of a construction domain must be proceeded by an establishment phase.</td>
<td>7</td>
</tr>
<tr>
<td>exploration,</td>
<td>Issues related to the exploration phase in Domain Construction Strategy</td>
<td>An exploration phase is necessary to construct the shared meaning of a coordination context.</td>
<td>13</td>
</tr>
<tr>
<td>immutable mobile,</td>
<td>Issues related to immutable mobiles. This is a concept in Actor Network Theory which indicates a network with strong properties of irreversibility and effects which transcend time and space, e.g. a standard.</td>
<td>The deployment may be hindered by immutable mobiles in the coordination domain. One example is the ClearDDTS tool for CR management at the A-domain.</td>
<td>4</td>
</tr>
<tr>
<td>resource,</td>
<td>Issues related to resources</td>
<td>Lack of resources made the deployment awkward.</td>
<td>10</td>
</tr>
<tr>
<td>trust boosting,</td>
<td>Issues related to the trust boosting phase in the Domain Construction Strategy</td>
<td>A constructed coordination domain must gain trust by using it in a sharp project before it is expanded to a larger user community.</td>
<td>18</td>
</tr>
</tbody>
</table>

The daily build and long term evolution described in Section 7.3.2 Evolution must be taken into consideration when deploying the Framework in an activity domain such as a development project. However, the step from applying the Framework in one project to several must not be taken too early. This was in fact what happened both in the S-domain and the A-domain:

“*And very soon decisions were taken [...] that other project should join in. So before we had a fully developed tool [...] did we enter a path which we didn’t intend to until after a while. We turned too soon and let the other projects join in way too early according to my opinion.*” (PM2-37:47)

The project manager PM4 suggested that a phased approach should be used to avoid these problems:

“*R: I think if you try to have at least per phase a stable base more or less. Because if you change specs during a specific phase the probably a lot of rework you have to do and you have to explain these changes to the ways of working, etc. I think if you can split it up per phase, prestudy, etc.*
From the outset the Domain Construction Strategy was merely a principle to be adhered to. The experiences indicate that an elaboration of the strategy is needed. This is done in Section 9.1.7 *The Domain Construction Strategy*, p. 221).

7.3.7 Data reliability

The consequence category ‘data reliability’ refers to the consequences for data reliability in the Information System. It is associated with the following effect categories:

Table 22. Consequence category – Data reliability

<table>
<thead>
<tr>
<th>Effect category</th>
<th>description</th>
<th>consequence</th>
<th>#</th>
</tr>
</thead>
<tbody>
<tr>
<td>champion,</td>
<td>Issues related to the champion role in the deployment</td>
<td>A prerequisite for achieving reliable data in eMatrix is that a champion is working in the project to help users load data into and extract data from the tool.</td>
<td>2</td>
</tr>
<tr>
<td>data consistency,</td>
<td>Consistency of data, e.g. no copies of the same data item</td>
<td>The data consistency has improved since all data are kept in one tool, eMatrix. Previously, several tools were used to hold coordination data items. Often the same item was stored in several tools which lead to severe data inconsistencies.</td>
<td>7</td>
</tr>
<tr>
<td>Matrix performance,</td>
<td>The performance of eMatrix in the Ericsson practice</td>
<td>The performance of the early versions of eMatrix was poor. This made users more reluctant to enter data into the tool.</td>
<td>16</td>
</tr>
<tr>
<td>Matrix data entry,</td>
<td>The loading of data into eMatrix</td>
<td>Project members were reluctant to load data into the tool. Not until the social pressure became too high did they do so (marking all missing data red in front of colleagues at project meetings)</td>
<td>16</td>
</tr>
<tr>
<td>Matrix stability,</td>
<td>The stability of eMatrix</td>
<td>The early versions of eMatrix were unstable. This made users more reluctant to enter data into the tool.</td>
<td>2</td>
</tr>
<tr>
<td>Matrix support,</td>
<td>Issues related to support</td>
<td>The on-site support at the S-domain was not sufficient. This made it more difficult for users to enter data into the tool.</td>
<td>7</td>
</tr>
<tr>
<td>Matrix user acceptance,</td>
<td>The user interface and the user acceptance of eMatrix in the Ericsson practice</td>
<td>The complex user interface of the web-client made it difficult for users to enter data into the tool. This was improved when the Lean Client was launched.</td>
<td>15</td>
</tr>
<tr>
<td>one management tool,</td>
<td>Issues related to one tool (eMatrix) to support coordination</td>
<td>Having all coordination data in one tool (eMatrix) improved the data reliability.</td>
<td>23</td>
</tr>
<tr>
<td>responsibility discrimination,</td>
<td>The pin-pointing of responsibility to the proper place.</td>
<td>The work package method and the increased precision in allocating responsibilities made it easier to see who was responsible for entering data into the tool.</td>
<td>14</td>
</tr>
<tr>
<td>trust,</td>
<td>Issues related to trust</td>
<td>The entering of data into the tool caught momentum only when actors felt that they could trust the data in the tool. One way of achieving this was to state that the only valid data are the data in eMatrix.</td>
<td>10</td>
</tr>
<tr>
<td>vendor,</td>
<td>Issues related to vendors</td>
<td>At the S-domain the consultants from the vendor had to be used for entering data into the tool due to lack of resources.</td>
<td>6</td>
</tr>
</tbody>
</table>
FRAMEWORK CONSEQUENCES

One place for all coordination information
When developing complex systems the maintenance of consistent and reliable data in the information systems is a major issue:

“R: But the next point is the most important one I think [having one information system for managing items rather then several], that is the one, that is the argument that has convinced the project managers and the head of the project office and the different people that took to use the tool, to have a consistent set of data.” (MT3-54:16)

One consequence of the Framework intervention is that the data consistency improved:

“R: I think that the Excel spread sheets they were maintained before in different places. They were inconsistent. They were always inconsistent. The technical coordinator has decided that an IP was removed but then in the others it still appeared. They had different titles and always, they had never the same revision.[...] during time they... This inconsistency that we found out when we tried to enter the information in eMatrix based on these three or four Excel spread sheets, then the [...] unintelligible] was totally lost because he had three different kind of information, which one was the truth...So I think the quality of the information has become much better.” (MT3-40:42)

Data entry - how to make it happen
A major difficulty in all coordination domains was to convince the actors of entering data into the tool (eMatrix). There were various reasons for this: the lack of support resources, the resistance to change established ways of working, the poor performance of early versions of the tool at distant sites, the problems to understand the user interface, etc.

One of the main reason was that the actors did not understand why they had to enter the data. From their point of view, entering data in a cumbersome tool was just an extra burden:

“R: We started to first put manually data into the tool from a central point. And then we asked the subprojects to put data into the tool as well. And the problem with the subprojects was that they did not fully understand how to do this. To some extent they did not understand why they had to do this, so it was always down-prioritized. What we ended up with was a sort of patch work information model... a patch work data in the tool meaning that we had bits and pieces that were more or less correct and others were not correct and had to be corrected.” (PM3-21:37)

However, there were ways to make the data entry happen. One strategy used in the A-domain was to expose the responsible persons in public for not having done their job:

“R: So taking the database as input for the discussions at the project meetings, that was the main driver for the subproject manager to put the information there. If there were reports were... all the Work Packages were red just because they didn’t enter the information, and this is presented in a project management meeting, then of course the next time they had updated it.” (MT3-07:03)

The key issue concerning data reliability is to promote the quality of the data to the level where the actors feel they can trust it. This can be achieved by creating a chain of dependencies among the actors in such a way that if someone fails to enter the data, the others will suffer:
“I(9:55) Did that [the entering of data] improve much when this decision to use the database for reports was taken? R: Yes, it was used by several roles we have in the project, and I think each area, for example the technical coordinator he has used eMatrix, and the test has used eMatrix. And as soon as different roles take the reports as input for their activities you must trust them. That is the driver to motivate all the people to enter the information into the tool.” (MT3-09:55)

Underestimating the complexity

A general experience from all sites was that the complexity in launching a global information system was underestimated, both by the deployment organization at Ericsson and by the vendor of the IS. In hindsight, after the first difficult years at the S-domain, PM2 expressed this in the following way:

“I: Do you think that we underestimated the complexity in this, it’s pretty much stuff we tried to do at the same time, changing work patterns, global distribution, large project...? R: Yes, I think we did that.... several of us.... I’m wondering if sometimes... well, how shall I put it,... we were told that this wasn’t such a big deal, no big stuff to make a change like this and that it should be smooth and quick and there I think we underestimated the whole.” (PM2-41:45)
7.3.8 The Information System - eMatrix

The consequence category ‘The Information System (eMatrix)’ refers to the consequences of using eMatrix as the Information System in the Framework in the Ericsson practice. This consequence category is associated with the following effect categories:

Table 23. Consequence category – The Information System (eMatrix)

<table>
<thead>
<tr>
<th>Effect category</th>
<th>Description</th>
<th>Consequence</th>
<th>#</th>
</tr>
</thead>
<tbody>
<tr>
<td>Infrastructure</td>
<td>Issued related to infrastructure like the Intranet, routers, PC’s, etc.</td>
<td>The infrastructure at Ericsson was not sufficient for the early set-up of eMatrix. Severe performance problems were experienced at Australia and other sites.</td>
<td>3</td>
</tr>
<tr>
<td>IS - eMatrix</td>
<td>The eMatrix product data management (PDM) system which was used in the Framework</td>
<td>By using eMatrix as the Information system the intentions of the Framework could be realized. This concerns in particular the experiential way of working due to the ease of making changes of the implementation in eMatrix.</td>
<td>30</td>
</tr>
<tr>
<td>Matrix functionality</td>
<td>The functionality of eMatrix</td>
<td>The basic functionality of eMatrix was a good platform for constructing coordination domains. Several advanced functions could be added as the project needs became apparent, for example report generators.</td>
<td>7</td>
</tr>
<tr>
<td>Matrix performance</td>
<td>The performance of eMatrix in the Ericsson practice</td>
<td>The poor performance of the early versions of eMatrix became a major problem at all coordination domains.</td>
<td>16</td>
</tr>
<tr>
<td>Matrix training</td>
<td>E. g. training how to use the eMatrix</td>
<td>The effort of training users was underestimated. Consequently the user acceptance was poor in the beginning.</td>
<td>12</td>
</tr>
<tr>
<td>Matrix development</td>
<td>Issued related to the evolutionary development mode of eMatrix in the Ericsson practice</td>
<td>The evolutionary way of implementing the properties of the coordination domain was successful. It provided more functionality faster at a lower cost as compared with other initiatives to provide support in a coordination domain.</td>
<td>14</td>
</tr>
<tr>
<td>Matrix reports</td>
<td>All kinds of reports from eMatrix</td>
<td>The report generators implemented in eMatrix was used heavily. This became the prime form for signifying dependencies. The navigational capabilities inherent in eMatrix was seldom used by casual users.</td>
<td>4</td>
</tr>
<tr>
<td>Matrix stability</td>
<td>The stability of eMatrix</td>
<td>The poor stability of the first versions of eMatrix caused severe problems. User reactions were harsh.</td>
<td>2</td>
</tr>
<tr>
<td>Matrix user acceptance</td>
<td>The user interface and the user acceptance of eMatrix in the Ericsson practice</td>
<td>The user acceptance of eMatrix has in general been low. The tool is considered hard to understand. Only with the advent of the Lean Client did this improve. In the Lean Client most of the particulars of eMatrix was hidden.</td>
<td>15</td>
</tr>
<tr>
<td>one management tool</td>
<td>Issues related to one tool (eMatrix) to support coordination</td>
<td>One important consequence of using eMatrix as the Information System in the Framework is that it is possible to implement the totality of the Framework models in eMatrix.</td>
<td>23</td>
</tr>
<tr>
<td>scalability</td>
<td>Issued related to scalability</td>
<td>It is possible to house both small and large projects in eMatrix.</td>
<td>1</td>
</tr>
</tbody>
</table>

The fundamental consequence of using eMatrix as the Information system was that the work package method could be used in the development of the complex Mobile Switching Centre node. As stated earlier, without tool support this would not have been possible.
Performance and stability
As described in Chapter 6 The history of the Framework the major problems with eMatrix were the performance and instability of the early versions of the tool. The poor performance was partly due to the insufficient capacity of the intranet at Ericsson for a tool like eMatrix. This state of affairs threatened the continued usage of the tool on several occasions:

“Well, in principle all the PA’s, or the subprojects, were enthusiastic using the concept because it had benefits for them as well. They could see the progress in other areas on which they were depending. However we had quite a lot of issues with performance. It simply was very slow outside of EED [the A-domain]. This was a major issue. People complained that just updating one attribute took them 2 hours or so. That simply takes too long. This came from several sites, Croatia, Australia, Dallas, wherever.” (PM4-02-3:00)

This situation was drastically improved in later versions of eMatrix and the introduction of the servlet based web client (called the Lean Client in the S-domain). Thus, a major lesson is that issues like performance and stability should be up front when deploying the Framework to new sites. Furthermore, the continued evolution of applications must not in any way jeopardize the performance.

Reports
A key success factor in the deployment was the report generators developed in the S-domain:

“R: I presented this way of working to them, to the mobile people. And they were really... oh this is nice and simple and easy to understand for everybody and very effective. In the end we had our online reports. People didn’t even have to go into the system themselves any more, they could just.... Especially the project managers, the information they needed directly from the web with one click in these pre-defined tables.” (MT4-19-2:05b)

These reports could present the information as straight tables or as tables displaying relational information. A particular generator called the Relation Viewer could display rela-
FRAMEWORK CONSEQUENCES

tional information in a matrix form similar to the dependency matrices defined by Eppinger et al. (1994). An example of the functional anatomy is given in Figure 50:

<table>
<thead>
<tr>
<th>PREL</th>
<th>READY</th>
<th>IN TEST</th>
<th>DELAYED</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C.P. definitions</td>
<td>function, C.I. definitions</td>
<td>function, D.E. definition</td>
<td>function, E.R. block / database</td>
</tr>
<tr>
<td>ON restart</td>
<td>function, D.E. definition</td>
<td>function, G.S. type def / print</td>
<td>function, START / restart</td>
</tr>
<tr>
<td>D.E. definitions</td>
<td>function, G.S. type def / print</td>
<td>function, J.I. LAB def</td>
<td></td>
</tr>
<tr>
<td>E.R. block / database</td>
<td>function, J.I. LAB def</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G.S. type def / print</td>
<td>function, J.I. LAB def</td>
<td></td>
<td></td>
</tr>
<tr>
<td>START / restart</td>
<td>function, J.I. LAB def</td>
<td></td>
<td></td>
</tr>
<tr>
<td>J.I. LAB def</td>
<td>function, J.I. LAB def</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 50. Dependency matrix, example

This kind of reports provides a good overview of the status in terms of states of functions being developed and the relations between them.

Useability

The usability of eMatrix was in general apprehended as poor, even by users who used the tool frequently:

“We also saw it in the PIs or the sub-project which were new; training was an issue. Especially in the early days there were not that many pre-defined reports and pre-defined queries. We had to do a lot in the tool itself and the tool as such is not that user friendly if I say it politely... I have been using it quite heavily but I still have problems in really understanding... the key concepts with it. That is definitely an issue... “(PM4-04-1:48)

This was however improved with the introduction of the Lean Client.
7.3.9 Efficiency

The consequence category ‘efficiency’ refers to the consequences of the Framework on efficiency in terms of quality, costs and efforts of performing certain tasks. It is associated with the following effect categories:

Table 24. Consequence category – Efficiency

<table>
<thead>
<tr>
<th>Effect category</th>
<th>description</th>
<th>consequence</th>
<th>#</th>
</tr>
</thead>
<tbody>
<tr>
<td>cost,</td>
<td>Issued related to costs, e.g., business cases. This has bearing on the possibilities for a new initiative to be realized.</td>
<td>Cost savings.</td>
<td>8</td>
</tr>
<tr>
<td>effort,</td>
<td>Issues related efforts, man-hours</td>
<td>Less efforts of performing tasks.</td>
<td>12</td>
</tr>
<tr>
<td>quality,</td>
<td>Issues related to quality</td>
<td>Quality improvements</td>
<td>10</td>
</tr>
</tbody>
</table>

Cost estimates

No quantitative cost analysis has been made and cost comparisons are difficult to do in a rigorous manner due to the problems of defining reliable cost measures. Furthermore, absolute costs cannot be discussed for confidentiality reasons. This means that any conclusion regarding cost effects must be at most tentative.

However, some relative analysis of cost effects can be made. As a reference we will take the overall costs of establishing the first coordination domain, the S-domain, between November 1998 and May 2000, that is, nineteen months. This cost includes the costs of Ericsson development personnel, consultants work and eMatrix licences. It excludes the costs of personnel working in the development projects. If we take the overall cost, normalize it to one year and put that cost to 1 unit, we get a figure of approximately 1.6 units for the overall period.

During these nineteen months the coordination domain as signified by the Conceptual Model in Figure 52 (p. 188) was constructed. It included

* requirement management,
* Change Request management,
* baseline and milestone management,
* test configuration management,
* support for incremental development including anatomy management,
* product and document management for the needs of the project,
* three general report generators: table reports, indented reports and matrix reports,
* an interface to the project document library,
* a thin, servlet based client (the so called ‘Lean Client’),
* global IS support comprising Sweden, Italy, Australia, Norway, Croatia among others,
* complete traceability between coordination items,
* on line status accounting, that is, monitoring of the status in the project,
* basic eMatrix functionality such as logging all events in the IS, defining user roles and access rights, security checks, etc.
FRAMEWORK CONSEQUENCES

The work was mainly carried out by myself, another Ericsson employee and 1-2 consultants and can be characterized as a long elaboration phase where the ‘daily tuning’ was employed heavily (see Section 9.1.7 The Domain Construction Strategy). In summary, we may say that the costs of establishing the S-domain with the properties above is 1 unit / year.

We may compare this cost with cost estimations concerning to the C-PDM system. For example, one implementation of a system to manage products and documents was estimated to take 0.4 years and cost 0.14 units which gives an estimated cost of 0.35 units per year (ERI-2000-03-17). However, the product and document part of the Conceptual Model are fairly stable. The other, more volatile coordination situations in Figure 44, p. 156 were not included. In fact, no implementation of C-PDM comprising all these coordinations exists at Ericsson.

Another comparison can be made with the cost of developing a requirement management application only in the C-domain in 2001. This cost was estimated to 0.9 units, that is, nearly the same cost as for developing the entire S-domain, including requirement management. The development was going to follow the standard linear project model, that is, analysis and prioritization of requirements, implementation prestudy, design and test in a serial sequence. Moreover, the number of actors involved in defining the requirement coordination situation were substantially increased. The consequences indicate that the experiential learning, which was applied in the S-domain, is more cost effective than the linear project model.

Similar indications of cost savings can be found in the data, for example:

“Especially during execution I think that in the old days we had a lot of manual administration for deliveries towards TCM [Test Configuration Management]. This has now been handled by the tool. [...]. I think that the tool is providing a lot of information which previously had to be obtained manually. That is I think saving cost now as well.” (PM4-8-1:15)

However, no decisive conclusions concerning cost effectiveness can be drawn from this study.

Effort

Also regarding efforts, that is, the man-hours spent on different tasks, some indications can be found in the data, for example:

“RM: I think that you manage to accomplish more tasks with the tool as compared with doing the same tasks manually. I think that you gain time on it, especially when you update CRs, take decisions on CRs and the like. I think that analyses are done faster now than before.” (CM1_2-29:40)

Another example is the time for doing status accounting:

“You can see quite clearly how the way of working has changed. Previously a configuration manager had to look up every document in a baseline in the project library, notice how far it had come, what kind of status it is in. Now he pushes a button and gets the current status on the whole baseline in a report, which also is pre-defined to display the data in the desired form, what data you want to include and so on. There you save a lot of time, that is one thing, the report generation is one of the things you save time most on.” (VDR1-078:45b)
However, no decisive conclusions can be drawn from the data regarding effort. On the contrary, the effort may equally well have increased for some tasks.

**Quality**

Regarding quality, there are some indications that the Framework contributes to an improved quality, for example regarding the location of errors sources:

“I: Did you have any idea about what kind of savings this has been in terms of time savings or anything else? R: [deep sigh...] No idea. The only thing I can say is we are now able to focus on the problems. We are not concerned with any more by simply rerouting request of information from one area to the other. There was a huge problem, we had problems enough. At least you had a good mechanism to pin-point where the problems were. I: Would you say that that has in some way contributed to the quality of the project... R: I think so, yes. Maybe not just due to the tool but the concept. Because you break up your work in small pieces. It makes it much easier to see where the problems are. In the good old days it was one big blob so to say. If there was a problem in there you had to really dive in and find out what the problem is. Now with the work packages it is pretty easy, or easy, relatively easy to pin-point where the problem is and then you can fix that part.” (PM4-06-4:04)

Another quality aspect concerns the interface between software and hardware parts of the system. Since these products are developed in highly specialized activity domains, there is always a risk that the interface is overlooked. Furthermore, if the isomorphism between the domains is not well established, there always the risk of misunderstandings between actors since their world views are so different. The following quotation indicates that the Framework contributes to an improved quality in this respect:

“RK: We saw earlier in the project where the hardware was [developed] in Italy that they took decisions in Italy based on the knowledge they had about the hardware and forgot that the software was affected. They did not implement the existing comments before the decision was taken. And that was devastating, we had to close a CCB [Change Control Board] which we had in Italy.[...] A change in the hardware could impact the software and they did not look into that. Here you can see this, here both the software and hardware in the projects go and look on each other’s stuff.” (CM1_2-23:55b)

Still another quality area concerns the consistency of data (see Section 7.3.7 Data reliabil-
ity, p. 174):

“R: I think that the Excel spread sheets they were maintained before in different places. They were inconsistent. They were always inconsistent. [...]...So I think the quality of the information has become much better.” (MT3-40:42)

Finally, there are some indications that projects, were the Framework was used, have performed well:

“The lead time of the project has not changed to the positive thanks to this... but we haven’t hit any serious mines which we previously did once or twice...”(CM1_2-24:49)
FRAMEWORK CONSEQUENCES

“R: Yeah, it is very difficult to compare that to previous projects. And there are so many different... issues that can impact the progress of a project, but... the project manager I spoke to he said ‘I have never run a project than went so smoothly as this one’ but I do not want to say that this is completely due to eMatrix.” (MT4-16-0:25)

7.3.10 Managerial issues

The consequence category ‘managerial issues’ refers to the consequences on managerial issues from the Framework. This category is associated with the following effect categories:

<table>
<thead>
<tr>
<th>Effect category</th>
<th>description</th>
<th>consequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>conflict,</td>
<td>Issues related to conflicts in the Framework context.</td>
<td>A number of conflicts resulted from the Framework intervention in the Ericsson practice.</td>
</tr>
<tr>
<td>credit,</td>
<td>Getting credit for a good job done</td>
<td>It is important to get credit for introducing the Framework in the practice.</td>
</tr>
<tr>
<td>entrepreneur,</td>
<td>Issues related to entrepreneurs driving an issue they believe in, often against established opinions</td>
<td>The introduction of the Framework presumes an entrepreneur who believes in the Framework and is prepared to drive this issue.</td>
</tr>
<tr>
<td>feedback,</td>
<td>Issues related to feedback on events</td>
<td>Actors in Australia did not get feedback on their proposals to improve the tool. This created a lot of frustration.</td>
</tr>
<tr>
<td>fun,</td>
<td>Issues related to having fun at work...</td>
<td>It is more fun to work with eMatrix than in the traditional way, especially for configuration management actors.</td>
</tr>
<tr>
<td>management support,</td>
<td>Issues related to top management</td>
<td>The introduction of the Framework presumes some kind of management support or ally. However, this does not have to be top management of the company. It may equally well be a project manager.</td>
</tr>
<tr>
<td>participation,</td>
<td>Issues related to actors participation in the activity.</td>
<td>The most vulnerable sites as for example Australia must feel that they are participating in the evolution of the Framework, otherwise the resistance to embrace the new tool will be low.</td>
</tr>
<tr>
<td>resource,</td>
<td>Issues related to resources</td>
<td>The resources were always tight during the deployment of the Framework.</td>
</tr>
<tr>
<td>trust,</td>
<td>Issues related to trust</td>
<td>If eMatrix and the data in it cannot be trusted, the users will resist using the tool.</td>
</tr>
</tbody>
</table>

Conflicts

The following conflicts (or rather contradictions) can be identified as a consequence of the Framework intervention in the Ericsson practice. Most of them are related in more detail elsewhere.

- **Rational programme:** Some parts of the Rational tool suite and processes were in conflict with the Framework IS eMatrix, while other parts, for example, ClearCase complemented eMatrix.
- **Corporate IS / IT:** The corporate PDM system C-PDM was the only PDM system recommended by Corporate. Since eMatrix was regarded as a PDM system, this caused conflicts with Corporate IS/IT.
- **Requirements prioritization:** Especially in the A-domain the prioritization of what should be implemented was a problem. During 2001 none of the requirements from
the network level was implemented.

- **The role of eMatrix:** This concerns what coordination situations should be included in the eMatrix scope. This conflict is discussed in Section 7.4 *Information System Architectures*.

- **eMatrix vendor:** The early versions of eMatrix had severe stability problems which caused a strained relationship between Matrix-One (the vendor of eMatrix) and Ericsson.

- **Management support:** Since eMatrix was not officially recognized, the line management at the S-domain was not actively endorsing the Framework initiative. One result of this attitude was constant arguments about resources to construct the S-domain.

- **Interpretation of concepts:** As the example in Section 7.5 *The construction of shared meaning* shows, the meaning of concepts was constantly debated.

- **Control of coordination domains:** With gaining organizational acceptance for eMatrix the fight for credit, influence and control of the evolution of the coordination domains increased. This was intensified during 2002 when Corporate IS/IT was given the control of the budget for the development of eMatrix applications in the C-domain.

- **Centralism - decentralism:** The contradiction between proponents for a federated, decentralized domain structure and the proponents for a centralized, one domain solution became eminent when the C domain was established.

- **Constructing the coordination domain or its elements in isolation:** In the S, A and L domains the ambition was to construct the coordination domains as a whole, that is, the models, methods and information system support were developed in concert with each other. In the C-domain the ambition was to develop the tools and methods separately.

**Entrepreneur**
By entrepreneur we mean someone who believes strongly in an idea, concept or innovation and has the ability to implement this in an organization. Unless such a person exists, the innovation may fail:

"R: I don’t want to mention it too open that this is really a success, because that... I mean.... it is not really sort of perceived and... the problem is that now the tool would need further development,... I left it with an initial implementation and there was a vision that I had at least to make it really work... But this is not driven by anybody. It is sort of halfway there and in the beginning it was a big improvement so they liked it but I’m very sure they will soon come to a point when they start hating it again... I: because of...? R: because they still have a lot of problems.... and then the debate will start again... maybe it was easier with the old tool. And if you don’t have a person who continues driving it,...” (PM3-83:39)

**Management Support**
Without some support from top management the innovation will not prevail:

"R: I think that one major problem is that there is nobody on the top who is driving it with a CM background. I: Do you think that is a major problem? R: I think that is one of the key problems, because if this is driven from somebody who just use it as any other tool, he wants to have a working tool and if he doesn’t get a working tool he doesn’t want to have any tool, then he is
looking for alternatives. But he has not the ambition to make this particular tool working. He will just have one set of solutions for his project. And I think you need to have somebody in the top team who is particularly focusing on making this work. And this must be their one task, then you can focus on improving this and maybe having smaller steps and stuff. But it takes time and it takes focus.” (PM3-28:14)

However, the management support need not be a line manager to succeed in the initial establishment. In both the S-domain and A-domain this support came from the project managers.

Resource allocation
Management needs to allocate enough resources to the domain construction work. This was not the case in the S-domain, which caused problems:

“R: Then it was the consultants, [management] wanted to spend as few hours as possible. At the same time they did not want to appoint more staff from Ericsson to the support in Beamon. So we got sort of what we had resources for. Well, this is the case with such decisions, [the line organization] has swayed a bit, and still does concerning what should happen with the tool and other stuff.” (PM2-43:00b)

Participation and feedback
The problems at the Australian site in the Beamon projects (described in Section 6.4.2 Fighting problems) could have been much alleviated if measures had been taken to counteract feelings of being left alone and not listened to:

“Yeah I hope there are some positives to be taken...it is only negative because the positive side is not coming through. It’s kind of like saying that this doesn’t work, and then someone says “OK here is the patch that fixes it”, then there is a big smile. The unhappiness here is that it doesn’t work but the fix doesn’t come through. And that’s where the frustration is, it’s not so much that tool is slow, it’s the fact that peoples kept saying it’s slow for so long. It’s not so much that the tool couldn’t do something, it’s that people kept saying it couldn’t do it for so long. And it just seems to take such a long time for things to happen. And we are very detached from the central expertise up there in Sweden. But I think people can see a lot of positives happening if the feedback is improved.” (MT2-44:48)

The issue of participation is further discussed in Section 8.3.1 Participation.
CHAPTER 7

7.4 Information System Architectures

A main issue all throughout the intervention of the Framework in the Ericsson practice was the scope of the Information System. Should there be one IS for an entire coordination domain or several? This is a question which borders to an entire complex of issues (see Table 26).

### Table 26. Consequence category – Information System Architectures

<table>
<thead>
<tr>
<th>Effect category</th>
<th>Description</th>
<th>Consequence</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>corporate IS/IT</td>
<td>Issues regarding corporate IS/IT</td>
<td>The information system architecture suggested by the Framework did not comply with the Corporate IS/IT strategy. One consequence of this was a number of conflicts.</td>
<td>7</td>
</tr>
<tr>
<td>global</td>
<td>Global aspects</td>
<td>The ISA used in the Framework was a central server with thin web-clients. This is a classical IRM architecture. However, this presumes thin clients for capacity reasons.</td>
<td>8</td>
</tr>
<tr>
<td>infrastructure</td>
<td>Issued related to infrastructure like the Intranet, routers, PC's, etc.</td>
<td>The infrastructure was insufficient for the ISA used in the Framework (the IRM architectures) until the Lean Client was introduced.</td>
<td>3</td>
</tr>
<tr>
<td>IS architecture</td>
<td>The role of eMatrix in relation to other IS</td>
<td>The management of all coordination items will be done in eMatrix. There are interfaces toward other IS, for example, ClearCase.</td>
<td>11</td>
</tr>
<tr>
<td>IS federations</td>
<td>Issues regarding one or several databases for eMatrix</td>
<td>A number of eMatrix environments may be arranged in a federal architecture which balances the need for central control with local autonomy.</td>
<td>16</td>
</tr>
<tr>
<td>local development</td>
<td>refers to local development of ADs (esp. tools) without coordination</td>
<td>Local development of a coordination is possible in a federated architecture.</td>
<td>8</td>
</tr>
<tr>
<td>one management tool</td>
<td>Issues related to one tool (eMatrix) to support coordination</td>
<td>The consequence of ‘one management tool’ is that the ISA will be structured in such a way that eMatrix will be the sole tool in the coordination domain.</td>
<td>23</td>
</tr>
<tr>
<td>separation of concerns</td>
<td>Categories related to hiding unimportant phenomena in a context</td>
<td>Separation of concerns is possible in a federated ISA.</td>
<td>20</td>
</tr>
<tr>
<td>stabilizing core</td>
<td>The stabilizing core of an activity domain</td>
<td>A federated ISA implies that there must be a stabilizing core which is mandatory and common to all federations.</td>
<td>4</td>
</tr>
</tbody>
</table>

Information system architectures (ISA) have been extensively discussed in the literature. In particular, the relation between the strategic planning of ISAs and business strategies has been discussed, see for example, Allen & Boynton (1991), Broadbent & Weill (1997), Earl (1996), Hugoson (1990), Reponen (1994) or Axelsson (1998). Here we focus on ISAs in relation to the coordination task, which is indirectly related to the business strategies through the system to be developed, manufactured and sold on the market. However, an analysis of the relation to business strategies is beyond the scope of this study.

#### 7.4.1 The traditional architecture

If we take the Conceptual Model shown in Figure 44 (repeated in Figure 51 below for ease of reading) the traditional ISA is based on the principle: one IS per coordination situation (the ovals in Figure 51). That is, one IS for requirement management, one for Change
FRAMEWORK CONSEQUENCES

Request Management, one for test management, etc. This is an information system architecture (ISA) found in many product developing organizations, including Ericsson.

There are several reasons for the emergence of this kind of ISA:

- There are ISs available on the market for a particular coordination situation. For example, RequisitePro from Rational and DOORS from Telelogic are well-known products for requirement management.

- There is no conceptual model which interrelates the different coordination situations. Remember that the Conceptual Model in Figure 51 emerged by using the Framework in concrete development situations, and that this model has a long evolutionary history. In the absence of a construction strategy like the one in the Framework it is hard to see how a significant model shall be defined for all situations. Thus, without an overarching unifying model, the ambitions are reduced to isolated coordination situations where the world-view of each tool sets the agenda for the interaction between the tools.

- At Ericsson the option of managing all coordination items in one IS has not been conceived as feasible. The problems of implementing and deploying an enterprise wide Product Data Management (PDM) systems have been staggering. Yet this concerns but one of the coordination situations in Figure 51, the products and documents coordination situation. Moreover, this situation is fairly stable since the definition of products and documents have been stable for a long period. Other coordination situations, for example incremental development, is much more volatile.

Some problems related to this kind of ISA are:

- A number of interfaces between the separate ISs must be implemented in order to achieve traceability across several coordination situations. Such interfaces are in general hard and costly to maintain, especially if the ISs come from different vendors.

Figure 51. Traditional IS architecture
CHAPTER 7

• If the interfaces between the ISs are not implemented, the information must be manually transferred between the ISs. This entails a great risk that the information becomes inconsistent. This was the case at the A-domain and is further confirmed in a recent study from another project at Ericsson (Crnkovic et al., 2003, chapter 8: “Case studies”).

• Each IS has its own user interface. Moreover, each IS has its own particular ‘world-view’, that is, an inherent set of concepts, notations and procedures, etc. Thus the entire coordination domain appears fragmented and heterogeneous to the actors.

• Changes which impact all coordination situations are hard to make because of the inherent world-views in the disparate ISs. All changes affecting level one in the layered model in Table 34 (p. 217) are of this kind, for example introducing a new enterprise wide attribute for products. This implies that the ability of the coordination domain to react to overarching changes will be low.

7.4.2 The alternative architecture

In the Framework approach the coordination domain is constructed by actors using the models and the Information System in the Framework as objects during the construction. This means that all the coordination items, which the actors perceive as significant, can be managed in one information system, in this case eMatrix. This is illustrated in Figure 44:

This idea became known at Ericsson as the ‘one-management-tool’ concept. It has some definite advantages in comparison with the traditional architecture:

• The interfaces between the ISs in the coordination domain are not needed.

5. A concrete example is an attribute holding export restriction codes for products.

188
FRAMEWORK CONSEQUENCES

- A number of IS can be replaced by one IS.
- The coordination information is consistent.
- There is a homogeneous worldview for the coordination domain.
- Changes which are common to all coordination situations are easier to do.
- The ability to react to imposed changes is higher.

The ‘one-management-tool’ concept was however not implemented to its full potential at Ericsson. Two main reasons for this can be identified. In spite of its apparent simplicity, the concept was in general not understood outside the group of actors involved in the construction of the different coordination domains. The other reason was that some existing ISs had acquired the state of ‘black boxes’ according to the Actor Network Theory (Latour, 1998). That is, their usage was already internalized and unreflected in the coordination domain. An example from the A-domain is the use of an IS from Rational\(^6\) in order to manage Change Requests although it could equally well have been done in eMatrix.

In spite of this, the results from the S, A and L-domains indicate that the ‘one-management-tool’ is feasible. The Conceptual Model in Figure 44 is an example from S-domain where all coordination items were managed in eMatrix.

The ‘one-management-tool’ concept does of course not exclude the use of other IS or tools needed to develop a complex system. For example, the management of files for software development should be done by special purpose software configuration management (SCM) tool such as ClearCase from Rational. This also means that there will be interfaces between the IS in the Framework and other ISs. However, these interfaces will not lie within the coordination domain. The issue of whether a certain item may be considered as an item in the coordination domain or not is ultimately a question about the character of the activity domains involved.

7.4.3 The corporate connection

The basic organization principle of Ericsson has been decentralization. Strong, independent local companies around the world have been given a large freedom in deciding about their processes and information systems support for product development. In short, only two mechanisms have been common: an enterprise wide legacy product archive where all released products are stored and a common set of rules for how to manage products and their describing documents\(^7\).

This arrangement has many advantages. However, one main disadvantage is that the local solutions may become incompatible. This is especially evident when it comes to archives for storing project documentation. Many local archives are used which creates a lot of problems, above all when the products are released to the common product archive:

---

6. ClearDDTS
7. A third mechanism is now being implemented: an Enterprise Resource Planning (ERP) system.
“[..] people were so bloody dissatisfied with Delta [a legacy document storage system] and the project archives on the design side and [this lead to] an investigation which showed what a chaos we had in the project archives.” (IT2a-42:42)

In order to bring some order into the project archives some local companies started to develop their own solutions based on the C-PDM system. This was done without any coordination between them. In 1998, Corporate IS/IT was given an assignment to come up with a strategy to consolidate these different solutions into a unified one. Thus, these Corporate initiatives were launched during the same period as the Framework initiative based on eMatrix began to catch momentum in the S-domain.

All the initiatives based on the C-PDM system were by and large confined to managing products and their associated describing documents, see Figure 53.

Thus the scope was the same as for the common product archive. However, the intention was to bring order in the project archives, not to replace the legacy archive. The project archives comprised more than 30 development sites and it was of outmost importance to consolidate these archives.

This means that both the C-PDM system and eMatrix originated in the needs of archiving project documentation which is contrary to most applications of PDM systems. The usual application area for PDM systems is exactly the area which the Ericsson legacy system covered, that is, product and document management and release archiving.

---

8. The PDM system that Corporate IS/IT endorsed.
9. Since the C-PDM system was based on a commercial, well established PDM system it was close at hand to consider this as a replacement for the legacy system, which was built and maintained in-house. It is however beyond the scope of this study to discuss these matters.
FRAMEWORK CONSEQUENCES

However, the main difference between the C-PDM application and the eMatrix one are the parts of the Conceptual Model that lie outside the oval in Figure 53, that is, requirement management, incremental development management, etc. This was included in the eMatrix applications but not in the C-PDM applications. This means that the architecture based on the C-PDM system in fact is of the traditional type in Figure 51. Consequently, it will suffer from the disadvantages of that architecture.

The consolidation of the different C-PDM applications was complicated by the fact that there was no shared meaning about what constitutes a product. An investigation showed that around 30 different product models existed at Ericsson. However, these models were believed to have a common core:

“R: Well, exactly, but that depends on that they went too far out towards the edge so to speak... [...] In the inner core there is no big discrepancy.” (IT2a-92:55)

This may be interpreted in the following way: the 30 different product models appeared in different activity domains. They shared a common core (albeit not articulated). However, in order to coordinate the development project, the scope had to be expanded into areas surrounding the oval in Figure 53, for example with Change Request handling. Since there was no shared meaning about the outside of the oval, different product related models were constructed in the different domains.

The consolidation of the local C-PDM applications was driven by Corporate from a firm conviction that products and product describing documents should be treated in the same way all over Ericsson.

“I don’t think that it should be possible to re-model the product core. That should be a basic model that everyone [...] sticks to [...]. We had in the back of our heads that we should have a, so to say, basic model that should be the same in the whole bloody company whatever you are doing. [...] We have argued that this core should be pretty large and that you should not keep on manipulating it.” (IT2a-65:02b, IT2a-66:53)

This was quite in line with the ideas in the Framework. A common stabilizing core is needed where enterprise wide mandatory elements are specified. However, while the eMatrix applications comprise the entire Conceptual Model, including the core product part, the C-PDM alternative comprises only the product and document parts. I claim that the decisive difference between the two alternatives is the ease of making changes in eMatrix as compared to the C-PDM system. This makes it possible to construct the entire coordination domain according to the Domain Construction Strategy in the Framework. This way of working is not possible in the C-PDM system.
CHAPTER 7

7.5 The construction of shared meaning

The consequence category ‘construction of shared meaning’ refers to the impacts from the Framework on the construction of shared meaning. This category is associated with the following effect categories:

Table 27. Consequence category – Construction of shared meaning

<table>
<thead>
<tr>
<th>Effect category</th>
<th>description</th>
<th>consequence</th>
<th>#</th>
</tr>
</thead>
<tbody>
<tr>
<td>communication,</td>
<td>Issues regarding communication between actors.</td>
<td>The Framework increases the communication among actors in the coordination domain.</td>
<td>9</td>
</tr>
<tr>
<td>experiential learning,</td>
<td>Issued related to learning by alteration between reflection and action. The iterative way of working is one example of this.</td>
<td>The experiential way of working in the Framework indicates that learning takes place among the actors which construct the coordination domain.</td>
<td>29</td>
</tr>
<tr>
<td>organizational learning,</td>
<td>Issues related to how the organization learns including evolutionary issues</td>
<td>An intended consequence of the Framework is that individual learning, shared meaning and organizational artefacts are constructed simultaneously. This may be regarded as an organizational learning concerning how to coordinate the development of complex systems. In this sense, the Framework contributes to organizational learning.</td>
<td>32</td>
</tr>
<tr>
<td>shared meaning,</td>
<td>Issues related to understanding, interpretation and meaning</td>
<td>The Framework contributes to the evolution of shared meaning.</td>
<td>38</td>
</tr>
<tr>
<td>signification,</td>
<td>In the Activity Domain Theory signs are mediators between the human cognitive system and the environment of the individual.</td>
<td>The models and the IS in the Framework may be apprehended as signs which mediate the relation between the individual cognitive system and the environment. This has consequences for the transcendence from the traditional way of working.</td>
<td>10</td>
</tr>
</tbody>
</table>

The intended impact from the Domain Construction Strategy is that shared meaning concerning the coordination domain emerges in the activity of constructing that domain. By working together in the domain, the meaning of phenomena will be determined by the consequences of the actions performed by the actors (see Wittgenstein, 1992/1953). There are several indications that this in fact happened, for example:

*R: I think that the traceability we used heavily. For we didn’t have one object type just being on it’s own, it was always to show impacts of relationships to other, and... what this traceability...when we talk about relationships in eMatrix, then there were different roles being responsible for the part and that together, all the different roles together, they build the overall picture then." (MT3-56:52)

7.5.1 The agony of achieving shared meaning - an example

The first conceptual models had evolved for several years in the work with the incremental development package and at the first sharp projects using eMatrix at the S, A and L-domains.

During 2001 a project was started to unite the different Conceptual Models into a common model for the C-domain. This was mainly discussed by actors in the core network area of Ericsson. During 2002 the radio area, who had been working quite separately from the core network area, began to take part in these discussions.
The following discussion took place on April the 11th, 2002. A meeting had been set up to discuss what a common conceptual model should look like. The purpose of the discussion was to agree upon a conceptual model for requirement management, a small but important part of the entire model. In the discussion the already established Conceptual Model from the S-domain was used as a starting point (see Figure 54).

A complete shared meaning of this model implies that the actors must agree on the meaning of at least the following items:

- The types of requirements,
- the states or life cycle of requirements,
- attributes on requirements,
- relations to surrounding items such as ‘requirement issuer’,
- attributes on relations,
- cardinalities on relations,
- revision stepping rules which state how relations are treated when new revisions are instantiated of a requirement,
- actor roles in relation to requirements, for example ‘requirement coordinator’ or ‘configuration manager’,
- access rights for roles, i.e. who is allowed to do what with requirements, for example create or delete a requirement.

Two different types of requirements (‘input’ and ‘detailed’) had been used in the S-domain for some time. One type originated from the customer (‘input’). The other type (‘detailed’) was the customer requirements translated to suitable requirements for product develop-
ment. The names of the requirements were well-known and established in the switching development domain but hardly outside that domain. The RMF (Requirement Management Framework) mentioned in the discussion was an attempt to define a common, tool independent requirement management policy for all of Ericsson. Telelogic is a vendor which a.o.t. sells the tool DOORS.

In the meeting representatives from both the radio area (A, B), the core area (E, F, LT) and C-domain developers (C, D, G) were present. LT is myself. The discussion was recorded with the permission of the participants (Mindisc 19, 2002-04-11). Here is an excerpt from the discussion:

- LT: All the time at UAB it was very useful to separate between these two types of requirements.
- A: Between what?
- LT: Between ‘input’ requirements and ‘detailed’ requirements, because they had been working like that for a long time and we couldn’t do without them.
- B: But isn’t it also that those input requirements are compiled from, I should say, from requirement from different issuers? Because that is also one thing to remember, that you compile several requirements into one requirement if there similar.
- LT: Yeah, that’s the task for the product management.
- B: We need to understand that too, that’s the thing that can happen at each product management level to do that kind of work.
- C: OK, yes.
- E: But one thing we need here is flexibility in use of the system, because we have different needs, different history, different practice in different organizations.
- LT: It would be nice to agree on something. If we can only agree that there is something called requirements then I think we are lost.
- B: But we can agree that we have customer requirements...
- LT: and design requirements...
- D: decomposed requirements and whatever.
- E: For me at the moment... on the level we are discussing right now we don’t care about this ‘detailed’ requirements, decomposed or system requirements. I know what it is, for me.
- LT: Isn’t it important to mirror that what you know in the model here?
- C: Yes, you have to have that in the prescription.
- LT: Or do you mean that you can take away these two types and still... or are they important to you?
- E: Yeah, they are important to me, yes. I mean, you have two different kinds of requirements.
- LT: Then I think we agree that we need two different types of requirements.
- G: We have two types of requirements.
- LT: Then it is just to change the names of those requirements.
- E: Do that if it is necessary, for me it is not that important.
- G: What it should be then is that this is a customer requirement.
- LT: Why not call it ‘stakeholder’ requirement? That’s what at least Telelogic calls it.
- C: [laughter]: Then we have another word for it!
- LT: So we can use DOORS [more laughter]
- [everybody are talking at the same time]
- LT: But change then ‘detailed’ requirement to ‘design’ requirement or system require-
ment or whatever so it has more of a connotation of a testable requirement.
- G: What I would like to do first of all is to get rid of this one, the parent-child relation
should only be on the ‘detailed’ side.
- LT: But if you change one of these relations you must change the other one as well.
- C: No but the stakeholder requirement must be connected to the ‘detailed’ requirement.
- G: It has to be a connection between them.
- C: How do you draw that then?
- G: I do it like this [shows on the board].
- C: Can you do that in the model? Is that possible?
- G: Yes.
- LT: Change the name of that to ‘detailed’ requirement before we go further we will be
lost here, you can’t have ‘detailed’ and ‘stakeholder’.
- B: It’s not the same.
- LT: I mean if you should change....you should change it on both levels.
- A: Actually I think I have problems with ‘systems’, ‘input’ and ‘detailed’, I don’t even
see the relation there, the logic.
- LT: What kind of names would you prefer then?
- A: [long pause] What does RMF say? [laughter]
- C: That is a good question, what does RMF say?.
- G: They don’t say that much.
- LT: They don’t say anything about this.
- G: But stakeholder... anybody can be a stakeholder... that’s wrong. A developer could be
a stakeholder.
- B: But not when he is inside the project.
- E: I think it is really good to have ‘stakeholder’, because a stakeholder can be a cus-
tomer.
- --------
- LT: What’s the reluctance of changing the name of the ‘detailed’ requirement?
G: I’m not reluctant, only that no one could say the name of it!
LT: Say ‘design’ requirement if that’s better.
G: I like that better.
B: But then what do we call the ones that we name ‘main requirements’ then? Because that is the same level as system level?
LT: We have requirements that can be tested and we have requirements that can’t be tested.
A: ‘Product’ requirements?
G: But it could be requirements on whatever.
E: Maybe ‘product’ is better.
C: But isn’t ‘stakeholder’ requirements put on products?
LT: But the point about ‘stakeholder’ requirements is that you can’t test them. We are going back again. You must have the freedom to formulate requirements that can’t be tested.
C: Give an example of a requirement that can’t be tested!
LT: “I want a car that runs fast”
C: You can test that!
LT: How?
B: That’s not a good requirement, you should say how fast.
LT: Exactly, you should say how fast and that is then the design requirement.
D: But it’s just an hierarchy I think.
LT: No, it’s not, it’s different.
A: But, maybe we need both [laughter].
G: Then we have that in the project preferences that we have a couple of requirements and that you can state whatever name you would like to be shown.
LT: But, let’s put it another way. These two types of requirements have been found good to have in the Ericsson practice and it has been found good to separate them. Can’t we just leave it like that?
F: We have to clarify that the first one is testable and the second one not testable.
A: I mean, the advantage to call it product requirement is that could be any type of requirement. It could be market, supply, it could be functional, it could be realization.
LT: Well then you have quite a different thing... but the only real difference I see is really that some requirements are expressed in customer terms, other requirements are expressed in design terms or project terms. And there is a need to separate these two, believe me.
G: Yes, I believe you and the thing is that we are discussing one thing in a model that has like 200 objects and it takes two hours. This is not the way to do it, it’s impossible.
FRAMEWORK CONSEQUENCES

Some observations
As G indicates above, after two hours of discussion, there was no consensus among eight actors about the meaning of one element (the requirement type) in a Conceptual Model which may contain several hundred elements. In addition to that, these elements will change due to external circumstances, new insights, new coordination situation, etc. It is quite easy to imagine the difficulties of a larger group of actors (say hundreds) to agree on the meaning of all the elements in the Conceptual Model.

Furthermore, there is no objective criterion by which a ‘correct’ model can be evaluated. There is no ‘correct’ answer as to which type(s) a requirement should have since this is completely arbitrary. Any proposal suggested above is equally valid in the same sense as a signifier of some signified phenomena in a social context is arbitrary. For example, the word ‘dog’ in the English language which signifies a quadruped with hairy fur which makes occasional sounds, is completely arbitrary.

Like Wittgenstein we believe that the meaning of a word is determined in the social activity among a group of actors (Wittgenstein 1992/1953). As long as the usage of the word in some action results in the intended consequences, the word is ‘correct’. When misunderstandings or disagreements occur, the meaning of the word must be considered and reflected upon. This means that the only criteria by which the discussion above can be resolved is the usability of the word. If the ‘Input’ type of requirement is useful in an activity domain it is ‘correct’. This position can be seen in some of my comments above.

This means that the disputes in the discussion above cannot ultimately be resolved around the table in the conference room. The Conceptual Model must gain its credibility on the combat field where projects are using it to develop products. Since the model is implemented in an information system, its usability is ultimately tested by the usage of the information system in the development practice. If the model is detached from the practice it will become a reified phenomenon without relevance. This is in fact what happened at Ericsson. The use of the Conceptual Model was detached from its implementation and usage in eMatrix and turned into a product of its own which is now maintained by discussion (see Chapter 6 The history of the Framework).
8 Reinforcement rods

The purpose of this chapter is to discuss research question RQ3: “Which elements in the Ericsson development practice contribute to successful outcomes of development projects? We will call such elements ‘reinforcement rods’ which is an allusion to the reinforcement rods used to reinforce the concrete in buildings. If such reinforcement rods are present in projects they will ‘reinforce’ the project ‘building’.

In order to find such reinforcement rods we will broaden our knowledge interest to projects within Ericsson where the Framework was not used (see Figure 55). We will also ground these rods in the Corporate IS/IT work with Product Data Management (PDM) systems. The rods are then compared with the findings from the Framework projects in order to find similarities.

Furthermore, from this comparison we can analyze how the Framework contributes to the ‘embedment’ of reinforcement rods in a project. Also, we may find reinforcement rods which the Framework does not address. Finally, there may be other reinforcement rods outside the Ericsson practice. However, as mentioned in Chapter 1 Introduction, it is beyond the scope of this study to identify such rods. This means that we do not claim that the list of reinforcement rods is inclusive. However, we claim that the reinforcement rods found are seminal for the outcome of the projects at Ericsson.

The analysis will proceed in the following way: We will do a fairly detailed analysis of the interview with PM1, who is a project manager with more than 25 years experience from Ericsson. This analysis will make use of the open, axial and selective coding in the Grounded Theory (GT) method (Strauss & Corbin, 1998), however somewhat simplified (see Section 2.2.7 Research processes, strategies and methods). The selective coding will be further grounded in the interviews of CU1, MT1, IT2 and DE1. The reinforcement rods will be separated into those that can be related to the Framework and those on which the Framework has no bearing.

8.1 The experiences of PM1

The interview questions are given in Section 2.2.2 Interviews. The project discussed in the interview was one of the largest and most complicated in the history of Ericsson so far. The task was to deliver a transit station to the customer, which was British Telecom in Great Britain. The size of the project, which was called BT Local, was around 800,000 man hours and it ran for 15 months between 1984-1985. The work was distributed on a number of so
called Local Design Centres around the world. PM1 was the total project manager for the project. The contract stipulated high fines for any delay of the deliveries.

The project was unique in many ways. The transit station was the first order from the British market and the project was important as a bridge-head for projects to come. The station in itself was not very large, only about 1000 subscribers, but it had a decisive strategic importance for the continued presence of Ericsson on the British market. Therefore, the project had a total support from top management, and every available resource was allocated to the project. The project participants worked on the limit of their capacity. Psychological problems and burnt out people were not uncommon.

The project was also the first one where the design was carried out at several development sites around that world. This put high requirements on coordination. The methods for development and project control were rudimentary and had to be defined during the course of the project. A large part of the staff was newly recruited and had no previous experience from development work at Ericsson. The technology was relatively new, and the software was written in an in-house language called PLEX (Programming Language for AXE). In summary, the project can be characterized as the first attempt at Ericsson to go from relatively small, local projects to large scale projects developed world wide.

8.1.1 Open coding

In the open coding I have read the transcript of the interview line by line and tried to identify concepts, categories, properties and dimension according to the GT approach (Strauss & Corbin, 1998, p. 101 ff.). Already in this phase it is possible to see that some categories are related to each other, something which according to GT shall appear during the axial coding phase. The number of categories is about 90. Most of them are well known and the open coding can be seen as a structured reproduction of the interview.

There are however some interesting phenomena which point to underlying, important categories. Take for example the fantastic event that took place during Christmas 1985:

“R: When Christmas time drew closer in 1985 we thought that many people had done a great job and I said as a project manager, well we should do something and in any case we sent a card to everyone, a Christmas card, but I thought that we should do something more, so at least a box of chocolates would be good. Said and done, we contacted Marabou [a Swedish manufacturer of chocolate boxes] and they helped us to send out these Alladin boxes of chocolate then. We used the list of people from kick-off meetings or introduction meetings, there we had lists of all the people on those meetings. However, the staff from the production units, well, they didn’t always think it was necessary to attend these meetings so we missed a number of people. And just around Christmas time when these boxes were distributed, we had a signal from the plant which were supposed to produce the hardware that the production line had stopped. They were on strike! And the head of the plant called and said, damn it, he said, everybody did not get these chocolate boxes so now they are sitting on the floor and do nothing. So I had to do a quick emergency job, back to Marabou once again and order them to send boxes of chocolate quick as hell and make sure that everybody get one. So that was an experience, whatever you do, DO NOT FORGET ANYONE, then you may have a backlash.” (PM1-27:31b)
This point to the reinforcement rod ‘participation’. Everyone in the project must feel that their work is appreciated and that they are part of the project. If this is not the case, the project may even be in jeopardy at times.

8.1.2 Axial coding

During the axial coding I have tried to separate categories and explanations expressed as conditions, actions and consequences (Strauss & Corbin, ibid., p. 128 ff.)

Categories

In the axial coding I have derived the following categories from the open coding categories:

- **MARKET**: categories related to the customer of the task.
- **PRODUCT**: categories related to the result of the projects.
- **TASK**: categories related to the execution of the task.
- **PROJECT**: categories related to the control of the task.
- **ACTOR**: categories related to actors in the task.
- **ORGANIZATION**: categories related to the organization of the business.
- **PROCESS**: categories related to the way of working when performing the task.
- **BUSINESS MODEL**: categories related to basic principles for the execution of the task.
- **KNOWLEDGE**: categories related to organizational knowledge.
- **COINCIDENCE**: categories related to unforeseen events during the execution of the task.

Again, most of these categories are well known within the development practice. Some categories, which are not usually recognized, are:

- **TASK**: Usually project and task are not separated. The reason why I have done so is that I want to see the task as the action which from the condition MARKET result in the consequence PRODUCT. The project is apprehended as the coordination and control mechanism for this action.

- **KNOWLEDGE**: The reason I want to highlight this category is that it is considered to be vital to the survival of the organization (Senge, 1990). It is apparent that many of the experiences made during this pioneering project were not transferred to later projects:

  “I: All these experiences made, did they stick in the organization or? R: No, this is so to speak a one-time shot, there is no guarantee... We are talking about a lead time of 15 months. It took us about 10 years to get back to this level again [...] I: It sort of vanished and then there was business as usual and then the ideas came back again? R: Exactly” (PM1-32:10a)

- **COINCIDENCE**: The unforeseen events that most projects are haunted by. One example:

  “Well, we missed the delivery date with one month, it was those bloody subscriber services that failed. The problem was known earlier, the interference between subscriber services. So unfortunately, the final test was that British Telecom came to Älvsjö where we in principle had
set up the Seven-Oaks station. And along they came with this little plastic card which they would deliver to their customers with these subscriber services, star and bla-bla, square and so on. After an hour they had sunk the station so it stopped. We had a fixing period of about a month.” (PM1-21:33)

Other examples are the previously related ‘chocolate box’ event and the following event that took place after the project. Top management turned from endorsing the project to take their hands from it:

“R: A final word, I’m sort of worried about this, was the project a success? Well, we made it apart from the troubles in the end. I wanted to arrange a big celebration for all the fantastic achievements done; there was a lot of overtime and extra work that everybody had to do. But then top management said: No this was not a success, you can’t have a celebration. I begged for some money for the organizations that had contributed so they could have a small party. But a larger celebration, that I didn’t get.” (PM1-44:08)

Explanations

In this section I extract from the interview a number of actions (A), the conditions (C) for the actions and the consequences (Q) from those actions. The overall action is regarded as the project (A) which shall deliver a product (Q) according to the needs of the market (C). The following main actions are found:

**Actions related to the MARKET:**

- The drive for the company to enter new markets (C) exercises an increased pressure (Q) on the project (A) and leads to an increased need for project management (Q).
- If the time plan is not followed, the customer may cancel the order (C). This increases the pressure on the project (Q). This in turn means that it becomes extremely important to have a correct picture of the status in the project (A) and that everyone in the project knows about the time plan (Q) and the importance of the fulfilment of it (Q).
- A picture, which is shared between customer and supplier (C) about what is to be delivered when (A), means that misunderstandings can be avoided (Q) and that the management of the project is simplified (Q).
- In this project the customer provided the time plan. This presupposes a stable market (C). This does not apply today (C) which means that the need for other development models in the project (A) increase (Q). This in turn implies that the need for coordination and control of the project increases (Q).
- The pressure of the market (C) implies that the system must be separated (A) into different loosely coupled parts (Q). This in turn implies two things: the possibility to deliver on time increases (Q) and that there is a risk that the system will degrade (Q).

**Actions related to the PRODUCT**

- A large amount of new technology (C) in the project (A) leads to increased management problems (Q).
- A group of experts (C), which scrutinize that the work of the designer (A) is in agreement with the basic architecture of the system, implies that the quality of the system can be maintained (Q).
REINFORCEMENT RODS

Actions related to the TASK
• The transfer of results between organizations (A) requires clearly defined result, a document (C). Otherwise, misunderstandings and delays may occur (Q).

Actions related to the PROJECT
• The introduction of more staff in the project (A) means that the need for support to manage the project increases (Q).
• Many new designers (C) in the project (A) implies that the need of coordination increases (Q). Moreover, they need to grasp quickly what their task is (Q). One way of doing this is to introduce simple work instructions (A).

Actions related to the ACTOR
• The actors must feel that they participate in the project (Q). This can be achieved by introducing a particular logotype which signifies the project and that everyone carries (A). This in turn leads to focused attention and feelings of belonging to a group (Q).
• BT local was a very important project for Ericsson (C). The importance of this is marked by a certain symbol which symbolizes feelings of status and importance. It is the task of top management and the project management to spread these feelings in the project (A). This in turn leads to a sense of belonging (“We are with you”) and this in turn leads to an improved performance from the staff (Q).
• If the basic prerequisites are not in place (methods, organizations, participation, etc.) (C) in the project (A), the pressure on the staff increases (Q) and many actors do not feel well.

Actions related to the ORGANIZATION
• The distribution of the development work onto several development organizations (A) leads to difficulties in coordinating the project (Q). This implies that the need for coordination support increases (Q).
• Top management support is a prerequisite (C) in order to carry out a large development project (A) when the prerequisites are as they were in BT Local (inexperienced staff, no previously tested development process, the expansion of the size of the project).
• An effort as large as BT local means that all available staff must be used (C). This leads to conflicts with other projects (Q). In order to get access to all resources in BT Local (Q) top management support is a must (A).

Actions related to the PROCESS:
• The usage of an overall process map (A) describes how things are related in the project. This leads to an understanding of the task (Q).
• The testing of the product is fundamental (A). One missed test case (A) may lead to undetected errors (Q) and delays in the project (Q).
• Working in all subsystems concurrently (A) will cause impacts of changes to hit everywhere (Q). In addition to this, it will be difficult to know what is valid (Q) and the coordination information cannot be held consistent (Q).
CHAPTER 8

Actions related to the BUSINESS MODEL
• Basic rules and principles (C) imply that the task (A) can be coordinated (Q).
• It is important to deploy the basics in the project to everybody (A). That makes everybody focus on the same goal in spite of differences in culture, language, etc. (Q).

Actions related to KNOWLEDGE
• Medium level line managers which feel left out (C) may counteract attempts to learn from experiences (A). The consequence may be that important experiences are not transferred to later projects (Q).

Actions related to COINCIDENCE
• If an unexpected event occurs (C), the management must act fast (A) in order to prevent the project from being affected (Q).

In summary, in the axial coding I did build on the results from the open coding and focused on a number of categories and explanations in relation to these.

8.2 Selective coding - reinforcements related to the Framework
So far, the analysis concerned categorizing and explaining “what is going on” in the study area on a fairly superficial level. The next step in the GT method is selective coding which according to Strauss & Corbin is “the process of integrating and refining categories” (Strauss & Corbin, ibid., p. 143). In doing so we used the axial and open coding in the proceeding sections.

The selective coding of the interview with PM1 resulted in the following reinforcement rods: Participation, Focus, Orientation, Federalism, Core. The analysis was done early during the research period (1999). Thus, this analysis can be seen as a first attempt to identify reinforcement rods at Ericsson. These rods have been refined and augmented with others during the continued research process. The Core reinforcement rod was renamed Stabilizing core and incorporated in the Balance rod. The additional rods found are Isomorphism, Evolution, Turning points and Shared meaning.

In the continued research process the selective coding was grounded in interviews both from contexts where the Framework was used and contexts where it was not used. For these interviews, an open and axial coding has been carried out. No detailed analysis has been made in terms of conditions, actions and consequences. However, the analysis of the additional interviews may be considered as a ‘theoretical saturation’ (Strauss & Corbin, ibid., p. 158) of the selective coding emanating from PM1. Besides the grounding in interview statements the reinforcements have also been grounded in internal Ericsson documents. Furthermore, for those reinforcement rods which are related to the Framework, the nature of this relationship is analyzed.
8.2.1 Evolution

In Table 30 the categories which we associate with ‘evolution’ are shown.

Table 28. Reinforcement rod – Evolution

<table>
<thead>
<tr>
<th>Category</th>
<th>Framework relation</th>
<th># non-FWK</th>
<th># FWK</th>
</tr>
</thead>
<tbody>
<tr>
<td>agile teams,</td>
<td>Issues related to small, independent teams without formal reporting obligations.</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>daily tuning,</td>
<td>Fast iterations between design and test</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>Domain Construction</td>
<td>The strategy in the Framework for constructing coordination domains</td>
<td>37</td>
<td></td>
</tr>
<tr>
<td>evolution,</td>
<td>Gradual changes of an AD</td>
<td>15</td>
<td>30</td>
</tr>
<tr>
<td>feed-back</td>
<td>Feed-back on events</td>
<td></td>
<td>14</td>
</tr>
<tr>
<td>IS - eMatrix,</td>
<td>The eMatrix product data management system which was used in the Framework</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>transcendence,</td>
<td>The dialectics between tradition and renewal</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>stabilizing core</td>
<td>The stabilizing core of an activity domain</td>
<td>37</td>
<td>4</td>
</tr>
</tbody>
</table>

|                       |                                                                                     | 54        | 123   |

The essence of this reinforcement rod is that new knowledge acquisition should be done gradually and build on an existing knowledge base. One reason why the AXE-S project failed may have been negligence of this reinforcement:

“Everything was built on untested technology, from telecom to the smallest detail in the software, that is, no stable fix-point existed. On the contrary, all the project parts did innovate new technology which in a miraculous way was supposed to work when put together.” (CU1-4)

Here is a couple of other examples:

“Within [the project] there were no established routines, methods or processes. The management was given full freedom which resulted in a start from ‘scratch’. Among the designers there were an evident spirit of ‘prototyping’ – they wanted to participate in developing new things by using new technology. This the one and only spirit was quickly diffused into all newly employed designers – since the communication was tighter between designers than between the management and the designers. This is a spirit which is very fruitful if you quickly want to test research results or new ideas – but it is not if you are going to develop world class real time products with high complexity.” (DE1-02)

“R: When I was employed at Ericsson I was a bit like that, young and hopeful and believed in large and fast changes and that you could do this and that. But L pointed out that ‘here we take small steps, more continuous, it’s about evolution and not revolution’. And I think that all these attempts to do a revolution we have seen, that is damn it..., well, that has not worked.” (PM1-62:07)
Thus, there must be some stable core in a complex situation to prevent chaos. However, it is a tricky balance to be upheld:

“R: If you can bring law and order, a common language combined with a flexibility and change when better things are available, then that is for sure a straight path to success. But the risk is that you will get stuck, you will get locked into this, we are incredibly document oriented at Ericsson at the same time as the way we define documents, it can mean bloody anything.” (IT2a-102:22).

8.2.2 Orientation

Orientation means that everybody has a common apprehension about what the project is all about and that each and every one knows her position in the whole complex. Ultimately this is about to get everyone to work in concert towards a common goal, to get the work to flow as effortlessly as possible. Furthermore, it must be clear to everybody what the consequences of actions, or lack of actions, are.

In Table 33 the categories associated with the reinforcement ‘orientation’ are shown.

<table>
<thead>
<tr>
<th>Category</th>
<th>description</th>
<th>Framework relation</th>
<th># non-FWK</th>
<th># FWK</th>
</tr>
</thead>
<tbody>
<tr>
<td>anatomy,</td>
<td>issues related to anatomies</td>
<td>Anatomies, which express dependencies between functions or increments, may be modelled in the Conceptual Model and implemented in the eMatrix.</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>dependencies,</td>
<td>all kind of dependencies, closely related to traceability</td>
<td>All kinds of dependencies, which are modelled in the Conceptual Model and implemented in the eMatrix, can be followed.</td>
<td>1</td>
<td>15</td>
</tr>
<tr>
<td>orientation,</td>
<td>issues related to orientation, see the whole picture, understand relations</td>
<td></td>
<td>7</td>
<td>9</td>
</tr>
<tr>
<td>overview,</td>
<td>overview of a complex situation</td>
<td>By changing focus between the detailed and the whole in the Information System an overview of a complex situation is provided.</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>traceability,</td>
<td>issues related to traceability</td>
<td>Traceability from any item to any item is provided by the Framework.</td>
<td>1</td>
<td>14</td>
</tr>
<tr>
<td>transparency,</td>
<td>issues related to transparency, to see how things are related.</td>
<td>The transparency in a complex situation is alleviated by focusing on small pieces of information at a time. This is provided by the capability of defining views in eMatrix.</td>
<td>14</td>
<td></td>
</tr>
</tbody>
</table>

It can be noted that ‘orientation’ is more salient in the Framework contexts. This should however not be interpreted as orientation is not important in non-Framework context, which, for example, the following quotations shows:

“When using the tools from the Rational Suite in development projects it has become clear that they lack support for management of certain data, especially within the management environment. [...] In the eMatrix environment information about all items can be managed in one tool: eMatrix. This means outstanding information management and traceability capabilities.” (ERI-2000-02-10, p. 16)

This quotation is taken from an internal investigation where the Rational tool suite was compared to the information system in the Framework, eMatrix. The quotation shows that
traceability is needed in the development practice, and that existing tools could not provide this.

Another example is the following:

“The lack of routines, established ways of working and business processes meant that we were unable to manage all the information which needs to be managed during the development of complex systems. Furthermore, we had extreme difficulties in relating information to its context. And without putting the information into context it is useless.” (DE1-04).

8.2.3 Federalism

By ‘federalism’ we mean, for example, that a development work (“the federation”) is divided into small, more or less independent areas (“the states”) in cooperation. The idea behind calling the reinforcement ‘federalism’ is to capture both the autonomous and coordinating aspects in one and the same concept. On the one hand it is impossible to develop a complex system without dividing it into smaller parts, which are somehow coordinated. On the other hand, coordination means restrictions which interfere with the autonomy.

In Table 30 the categories which we associate with ‘federalism’ are shown.

Table 30. Reinforcement rod – Federalism

<table>
<thead>
<tr>
<th>Category</th>
<th>Framework relation</th>
<th># non-FWK</th>
<th># FWK</th>
</tr>
</thead>
<tbody>
<tr>
<td>anatomy, contextuality, focal change, focus, IS federations, isomorphism, one management tool, responsibility discrimination, stabilizing core, work package</td>
<td>The anatomy is a mechanism to divide a complex system in smaller parts. The Conceptual Model signifies the context of the coordination domain. Focal changes is a way to switch between parts and whole in a federated structure. Each autonomous part in a federation should have a focus. Increments are a form of federalism It is possible to envisage a set of coordination domains which all have their own eMatrix implementation and still can exchange data by adhering to a common, stabilizing core. Different world-views should be one criteria for separating a complex situation in to smaller parts This indicates that the coordination domain should be considered as one concern. The pin-pointing of responsibility to the proper place. The Stabilizing Core in the Framework provides the mechanism to balance between central control and autonomy in a federation. Issues related to the work package method Work packages are a form of federalism</td>
<td>37 14 3 5 3 4 2 16 4 6 23 14 37 4 9</td>
<td>7 14 3 5 3 4 6 16 7 14 4 9</td>
</tr>
</tbody>
</table>

Federated structures can be found in the product, in the organization and in the development process. The product should be structured in such a way that it is possible to work independently with separate parts (modules). This mean among other things that the interfaces...
between the modules must be well defined and that the separation into modules should be
driven by the so called ‘module drivers’ (Erixon, 1998; Erixon et al., 1994). The organiza-
tion should mirror the structure of the product in the sense that related modules should be
within the same organization, possibly at the same geographical place. The development
process signifies how the coordination of the development activities should be managed. All
these aspects can be associated with ‘federalism’.

As can be seen from the many references to the data, federalism is a conspicuous reinforce-
ment, both in non - Framework and in Framework contexts. One example from the non -
Framework context is the following. The development organization had done a smart design
without consulting the production organization:

“Then I came to the plant in Norrköping to interview them about their problems [...] and then he
said that AKK-50, that bloody system, do you know how many straps we must insert in the at the
station in order to start a central? 2500, that means that we are designing the system each time we
install it... In short, there these people on the design side thought they had done a real slick design
and then the whole management chain was in a mess.” (IT2a-06:40)

An example from the Framework context is the following:

“And also based on the anatomy chart, which is also I think a very important mechanism, you see a
lot of dependencies both time wise and product wise, etc. So you can fix things that need to be fixed
first and fix problems that are at the end of the project so to say at a later stage. Of course that is
also a key issue.” (PM4-07:0:37)

8.2.4 Balance

‘Balance’ refers to the balance between order and disorder in an activity domain. At Erics-
son this concerns mainly the balance between what should be centrally controlled by Corpo-
rate and what should be left to the local autonomy of decentralized organizational units.

In Table 31 the categories which we associate with ‘balance’ are shown.

<table>
<thead>
<tr>
<th>Category</th>
<th>description</th>
<th>Framework relation</th>
<th># non-FWK</th>
<th># FWK</th>
</tr>
</thead>
</table>
| adaptation,            | The adaptation of the tool, processes, mod-
                        | els, etc.                                              | The ability to adapt is depending on the balance.       | 3         | 5     |
| balance,               | The balance between central control and local   |                                                         | 23        | 3     |
| centralism -          | Centrally versus decentralistic tendencies in an organization | Concerns the balance between centralism and decentralism. | 24        | 3     |
| decentralism,          |                                               |                                                        |           |       |
| flexibility,           | The ability to align the properties of an activity | The ability to adapt to changes is depending on the balance. | 1         | 9     |
| IS federations,        | Issues regarding a federated information sys-
                        |tems architecture                                       | A federated IS architecture needs balance between order and disor-
                        |                                                          | dier.                                                            | 2         | 16    |
| stabilizing core,      | The stabilizing core of an activity domain     | Balance is a functions of the stabilizing core.         | 37        | 4     |
|                        |                                               |                                                        | 90        | 43    |
It can be seen that ‘balance’ is more dominant in the non-Framework data. One reason for this may be that there has been a tendency for the Ericsson organization to oscillate between the two extremes of centralism and decentralism. This may reflect a higher sensitivity to the balance issue than in the Framework context, which was very much characterized by a pioneering spirit. Also, the difficulty I had in promoting a federated IS architecture, and the centralistic tendencies in the last phases of the Framework history, are signs of this.

8.2.5 Isomorphism

‘Isomorphism’ was defined by CU1 in the following way:

“R: I believe very much in that creating understanding is about creating isomorphisms, that is mappings between the world I do not know and the world I know. You expect that everybody knows about programming language concepts, and he who comes with a solution, he describes the isomorphism between what I have in my concept and what you know, the conceptual tools you know, classes and operations, pointers, etc. And if you do not have this, then you must have something else, but somewhere you need to have a common conceptual world, and then it is just about describing the new in terms that everyone can have as a frame of reference. And this is the purpose of an architecture, which must be translated into some concepts which are supposed to be well known among the designers.” (CU1-38:35)

In Table 32 the categories which we associate with ‘isomorphism’ are shown.

<table>
<thead>
<tr>
<th>Category</th>
<th>description</th>
<th>Framework relation</th>
<th># non-FWK</th>
<th># FWK</th>
</tr>
</thead>
<tbody>
<tr>
<td>shared meaning</td>
<td>Issues related to understanding, interpretation and meaning</td>
<td>Isomorphism presumes a shared meaning concerning towards which the mapping between worlds can be done.</td>
<td>43</td>
<td>38</td>
</tr>
<tr>
<td>stabilizing core</td>
<td>The stabilizing core of an activity domain</td>
<td>A stable element, for example an architecture, is needed if a mapping shall be possible.</td>
<td>37</td>
<td>4</td>
</tr>
<tr>
<td>isomorphism</td>
<td>Mappings between world views</td>
<td></td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>84</td>
<td>48</td>
</tr>
</tbody>
</table>

In short, isomorphism can be defined as a mapping between different world-views. Isomorphism can be directly related to the coordination constituent ‘domain translation’ and the Transition Model in the Framework. In addition to this, isomorphism presumes a stable frame of reference. This means that we may also relate isomorphism to the stabilizing core constituent and shared meaning.

The reason why we consider ‘isomorphism’ as a reinforcement is that we consider it fundamental for developing complex systems. In fact, the lack of isomorphism may have been another reason why the AXE-S system failed (Section 6.1 A pattern emerges (1990-1995):

“There was too large a distance between choices in the system architecture and the choice of software technology / architecture. This distance was ‘patched’ with a number of in-house developed languages. Furthermore, these could not be automatically translated to one hundred percent. The compilers generated skeletons of code which had to be managed manually.” (CU1-2)

Another example is the following:
“Just because we were a young team there were misunderstandings occasionally – especially within design. The designers were a closely united team – and that is good for the motivation and commitment – but there was no one from the management in the team. This resulted in a dramatic difference in values between the management and the designers. This had been amplified by the ‘prototyping spirit’. The consequence has been that several ‘micro-cultures’ are flourishing.”

(DE1-08)

8.2.6 Shared meaning

In Table 33 the categories associated with the reinforcement ‘shared meaning’ are shown.

Table 33. Reinforcement rod – Shared meaning

<table>
<thead>
<tr>
<th>Category</th>
<th>description</th>
<th>Framework relation</th>
<th># non-FWK</th>
<th># FWK</th>
</tr>
</thead>
<tbody>
<tr>
<td>communication,</td>
<td>issues regarding communication between actors.</td>
<td>The work package method implementation in eMatrix brought about an increased</td>
<td>9</td>
<td>4</td>
</tr>
<tr>
<td>experiential</td>
<td>learning, issues related to learning by alteration between reflection and</td>
<td>The Domain Construction Strategy contributes to the emergence of shared meaning.</td>
<td>11</td>
<td>29</td>
</tr>
<tr>
<td>organizational</td>
<td>learning, issues related to how the organization team including evolutionary</td>
<td>Learning about the coordination of the development of complex systems is crucial</td>
<td>4</td>
<td>32</td>
</tr>
<tr>
<td>shared meaning,</td>
<td>issues related to understanding, interpretation and meaning</td>
<td></td>
<td>42</td>
<td>38</td>
</tr>
<tr>
<td>signification,</td>
<td>signs are mediators between the human cognitive system and the environment of</td>
<td>The models and the information system in the Framework is apprehended as signs</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>the individual.</td>
<td>which signify relevant phenomena for the actors in the coordination domain.</td>
<td>70</td>
<td>113</td>
</tr>
</tbody>
</table>

As can be seen, there is a fairly high fraction of grounding data in non-Framework contexts. Especially, for the core category, ‘shared meaning’ the grounding in non-Framework data is high. Thus, shared meaning may be interpreted as a reinforcement which is general in the Ericsson practice and not specific for the Framework context. An example of this is the following quotation:

“Personally I believe that what hit us hardest due to the lack of business processes was that we spoke different languages – we had no common terminology. This concerned requirement analysis, system design, architecture, testing and all the way to product and delivery structure. Even single individuals applied and used concepts in different ways. Some of the sublime concepts were Incremental development, Network Use Cases, Data Model, Product Structure and finally a real treat: Sales Object – a concept which we could never agree upon.” (DE1-05)
8.3 Selective coding - other reinforcements
This section concerns reinforcements which are not directly related to the Framework.

8.3.1 Participation
At several occasions, PM1 emphasizes how seminal it is to get everybody to feel that they are participating and important, no matter what role they have. If you can succeed in this, people will perform beyond the ordinary. It is about showing interest and communicating with all involved. An important element is some symbol, a ‘sticker’, which marks the belonging to the group. Another element is that the management clearly shows that they support the project.

There are several studies which show that attention and a sense of being important raises the performance, for example, the famous Hawthorn experiments that Mayo did in the 1920s. So, the reinforcement ‘performance’ is important to the project success. It is also likely that a sense of ‘being with’ makes the organization more robust towards unforeseen events since the actors are less dependent on management for taking decisions.

Here is a good example of the importance of ‘participation’:

“R: It was a switching project that consisted of several small units, sub-projects, and usually the large and most important units get the attention. But we had a very good project manager R who cared about the small ones as well. The smallest one concerned a correction which we needed from GSS [an internal Ericsson acronym] I think it was. And in practice there was one guy who did this, he had nothing else to do, that was it. But R spent time and visited him in person, he was down there, and... well, after that he had no problem in getting his full commitment.” (PM1-38:30)

8.3.2 Focus
By ‘focus’ we mean that all necessary resources are aligned towards a clear and concrete target: what should be delivered at a certain point in time. Top management must give support and permission to transfer resources from ongoing tasks to the project in focus. This may have the effect that those who are not participating in the project become frustrated and wants to obstruct the business in different ways. For example, the resistance may hinder new knowledge to become institutionalized in the organization.

An example of the ‘focus’ reinforcement is given in the following quotation:

“Well, we were going for the Japanese market and then it was this time that had priority number one: That went beyond anything else. It could cost anything, it had to require anything and it, it could almost work in any way, well it should work pretty well but it the time above anything else that mattered. And we did succeed, we went live if I remember correctly the first of April 1994.” (MT1-19:25)

The target for the focus should be evident, if possible by having a simple and effective sign for the target. In this project the target was symbolized with a Japanese flag with the word “THE PRODUCT” in the middle of the red sun symbol.
An example from the Framework context is the following:

“I: Ok, this workshop this was backed by the top management... R: Yes, because one of the main goals of this workshop was that... we had our delivery dates, the ADs¹ agreed. We had an anatomy agreed. The final result of the workshop should be that we could present our anatomy plan in eMatrix. So we could just put it on the overhead screen and say: Ok these are the assignments that you have, that is what you will deliver to which AD...” (MT4-05-4:33b)

8.3.3 Turning points

‘Turning points’ refer to unexpected events in a program² which have a decisive influence on the future trajectory of the program. There are several examples of turning points in the empirical findings, for example:

• My ‘unforeseen meeting’ with the vendor of eMatrix at a fair in 1996 (see Section 6.2.2 The Information System enters the stage).
• The signing of the first set of eMatrix licences in 1998 (see Section 6.2.3 The pilot projects).
• The coincidence of the elaborated CPLtool with the needs of the Beamon project (see Section 6.4.1 The Beamon project).
• The new servlet based web-technology which enabled the development of the so called ‘Lean Client’ which disarmed the threat of stopping the entire program because of the performance problem experienced during 1999 (see Section 6.4.2 Fighting problems).
• The Corporate agreement with Matrix-One³ in 2001 which established eMatrix as a Corporate concern (see Section 6.6.3 Corporate on stage).

From the non-Framework context we have the previously mentioned priceless event with the forgotten chocolate boxes (see page 200) or the sinking of the BT Local transit station by just randomly pressing buttons (see page 201).

It lies in the nature of turning points that these cannot be foreseen. However, those responsible for the project must recognize the turning points and react quickly in order to promote the program or preventing it from collapsing. A precious example from the field is the following:

“Once I got a report from some guys which I believe were in Morocco or Algeria, it was so damn funny. They told us what a hell they had there, they didn’t get the right kind of screwdrivers and every morning, it said, they had to go down to the local market to buy back the relay-set which had been stolen during the night.” (IT2a-18:49)
In Chapter 9 **Discussion of the Framework Approach** the impacts on coordination from the Framework intervention in the Ericsson development practice are discussed as well as the Framework itself and its elements. Other consequences related to coordination are analysed such as the balance between central control and autonomy in an organization, the development of large, global information systems, the usage of one information system for the entire coordination domain and the construction of shared meaning. The results are related to the theoretical grounding of the Framework in the Activity Domain Theory. The generalizability of the results is also discussed. This chapter provides an answer to the research question 4.

In Chapter 10 **Conclusions** the conclusions from the study are given. The research design and methods are evaluated. Further areas of research are suggested. The chapter closes with some remarks concerning the quality criteria and limitations of the study as well as the significance of the study for Ericsson and other industries. Finally, some reflections on being an actor both in the industry and research community are given.
9 Discussion of the Framework approach

In this chapter we will discuss the results from Chapter 8 Reinforcement rods, Chapter 6 The history of the Framework and Chapter 7 Framework consequences.

9.1 The Framework and its elements

In this section we discuss the Framework, its elements and their interactions. We start with some reflections on the Framework as a whole.

9.1.1 Reflections on the Framework as a whole

The Framework has passed a number of stages during its trajectory at Ericsson. In the early 1990s it started as a vague conception where some elements were more expressed than the others. For example, the Conceptual Model was expressed early in the Incremental Development Package.

The Framework as a coherent whole was however not established until the Information System (eMatrix) and the experiential learning strategy were included and operationalised in the development practice. Only when the individual elements were brought together as in the S, A and L-domains, the full potential of the Framework became evident.

With the establishment of the central C-domain during 2001 and 2002, the Framework was once again split up into individual elements and the experiential learning strategy was abolished. The Framework ceased to exist as a coherent whole and the dialectical relations between the elements were broken. From that point on these elements were developed in isolation. For example, eMatrix turned from being the Information System in the Framework into an information system (IS) among others. This meant that focus shifted from constructing a coordination domain into constructing an IS. In the empiri, there are indications that this slowed down the continued evolution of the eMatrix application. For example, VDR1 compared the effort laid down in the C-domain with the effort in the S-domain in the following way:

"R: In practice, if you look at the functionality which has come out of the system it is incredibly little in a long time as compared with how much functionality we produced with two persons in half a year..." (VDR1-102:44)

This does not mean that the central, one-domain approach is without merits. On the contrary, there are a number of advantages with this approach, such as reduced number of domains to be maintained, a common conceptual model, a common process model, etc. However, it remains to be demonstrated that this approach is feasible and that the effort to agree on commonality across all development sites across Ericsson is not prohibitive. Moreover, things change, including commonality. A common domain is inherently inert to change since new knowledge gained on the ‘battlegrounds’ have a longer way to travel before they can be internalized as ‘common’ in the organization.
I claim that the major reason behind the fast evolution of the S, A and L-domains is that the Framework focus is on assisting the construction of the coordination domain as a whole and not individual element such as the IS. This makes it possible to attack the notoriously difficult problem of achieving shared meaning among the actors and at the same time promote information system support for the coordination task.

A consequence of this is that the constructive approach includes the construction of meaning and not only the artefacts in the Framework. This means that the models and the IS should be regarded as composite signs which signifies the coordination domain for the actors. In Chapter 6 The history of the Framework several signifying processes can be identified. A particularly illustrating example is the signification of the concept ‘increment’. Before the Incremental Development Method was elaborated in 1996-1997 the signification took place in the language game among the actors. People engaged in lengthy discussions about the meaning of ‘increment’. As long as the signifier of the sign was merely utterances, the signification process converged only slowly (see Section 6.2.1 The Incremental Development Method Package). The signifying precision of the sign was too vague. When the Conceptual Model was introduced in the method package, the signifier was amended with pictures showing ‘increment’ in relation to other concepts. The signification then became precise enough for the meaning of ‘increment’ to converge in a small group. However, the closure of the signification process came about only when the signifier was further amended with the signs in the information system (icons, browsers, cues, tables, text, etc.). The inclusion of the IS in the signifying process made it possible to directly try out the consequences of a particular interpretation in action. Thus, alluding to Wittgenstein (1992/1953), if the consequences of a certain action were the anticipated, a shared meaning of ‘increment’ had been achieved.

9.1.2 The Conceptual Model

The Conceptual Model signifies the structure of the coordination domain in terms of coordination items and their static relationships.

A number of examples of the Conceptual Model are given in Chapter 6 The history of the Framework. These models have an interesting interpretation in terms of ‘organizational memory’:

“Information interpretation is greatly affected by cognitive maps or frames of reference, which are undefinable except in terms of a memory.” (Huber, 1991, p. 107)

Organizational memory has, according to Huber (ibid.), a human component in addition to artefacts in the organization. The models in the Framework reflect the shared meaning of a group of actors at a particular point in time and space. This means that the evolution of the models can be interpreted as the evolution of the organization’s memory. Thus, the models can be seen as concrete pictures of how the organizations ‘memory’ has evolved for a particular domain.

The conceptual models shown in the examples are defined at a fairly high level of abstraction. In order to implement these in the Information System they have to be more detailed.
DISCUSSION OF THE FRAMEWORK APPROACH

Early 2002 I suggested the layered structure shown in Table 34 for the Conceptual Model (LTX-2002-03-28):

Table 34. Layers in the Conceptual Model

<table>
<thead>
<tr>
<th>Layer</th>
<th>Purpose</th>
<th>Characteristics</th>
<th># of elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Trans coordination domains</td>
<td>Understanding, communication</td>
<td>Mandatory and stabilizing across a set of coordination domains</td>
<td>~20</td>
</tr>
<tr>
<td>2 Coordination domain specific</td>
<td>Understanding, communication</td>
<td>Coordination / management within a coordination domain</td>
<td>~50</td>
</tr>
<tr>
<td>3 Coordination domain specific detailed</td>
<td>Understanding, specification for implementation</td>
<td>Detailed rules for coordination / management</td>
<td>~100</td>
</tr>
<tr>
<td>4 Coordination domain specific IS dependent</td>
<td>IS implementation</td>
<td>Specific for each tool. Details found in IS only</td>
<td>&gt;1000</td>
</tr>
</tbody>
</table>

The first layer was meant to contain only mandatory and stabilizing elements across Ericsson as a whole. The main purpose of this layer is understanding and communication, that is, its signifying properties are in focus.

The same goes for the second layer. This layer is specific for a particular coordination domain and inherits the feature of level 1. Also in level 3 understanding and communication are in focus. However, on this level the model is specific enough to be implemented in eMatrix, that is, on this level the details of the coordinate domains are shown. Typical features on this level are cardinality on relations and revision stepping rules. The final level is visible only in the IS. The number of elements increases as the level of detail increases.

The importance of the signifying aspects of the Conceptual Model was difficult to disclose. Some actors argued vigorously for using the UML\(^1\) notation, which I opposed. The argument was that UML is a de facto standard, which is used outside Ericsson as well. However, most of the actors that participated in the construction of the coordination domain, were not acquainted with UML and its rich formal notation. Thus the signifying process is more difficult in comparison with the informal notation used in the Framework. This was indirectly confirmed by the return to the informal notation at the C-domain after some attempts to use UML.

9.1.3 The Process Model

The Process Model signifies the dependencies between activities impacting the coordination items. Thus it corresponds to the definition of coordination according to Malone & Crowston:

“Coordination is managing dependencies between activities” (Malone & Crowston, 1994, p. 90)

In the Framework the notation of Information Flow Diagrams (IFD) was used. One reason for this is the ability of IFDs to signify the relation between the orientational and temporal

---

1. UML: Unified Modelling Language
coordination constituents. IFDs were used to model hardware development processes in the AXE-S project, in the Incremental Development Method Package and in the definition of the Change Request process in the S-domain.

As compared with the Conceptual Model, the Process Model in the explicit form of IFDs has played a minor role in the Framework. However, the implementation of state chains in the form of policies in eMatrix is evident. There is a relation between policies and IFDs since each occurrence in an IFD, where an upward arrow meets a line, corresponds to a state transition in the policy.

One reason why the IFDs are hardly used at all in the Framework, is that the Conceptual Model, the policies / state chains and the anatomy (see Section 7.3.2 Evolution, p. 161), when implemented in eMatrix, are sufficient to manage the dependencies between activities. Also, the development of a product such as the Mobile Switching Centre node (see Figure 2, p. 3) cannot easily be captured in an IFD since design work is carried out in many work packages simultaneously.

9.1.4 The Transition Model

The Transition Model signifies how the coordination domain interacts with other activity domains, that is how coordination items are interpreted and translated when they appear in other contexts.

The Transition model was used early in the AXE-S project in the form of the Specification Based Data Model of Gandhi & Robertson (1992). In the Framework two main usages can be identified: in modelling the anatomy (see Section 7.3.2 Evolution, p. 161) and as a ‘sensitizing instrument’ when interfacing the coordination domain to other activity domains.

As described in Section 5.1.3 The Transition Model the essence of the Transition Model is a recursive specification – implementation structure on which the scope of each increment can be mapped. This is illustrated in Figure 56a where two increments are defined. The functional anatomy is in essence this structure with the implementation part subdued, that is, a specification-needs-specification structure (see Figure 56b). In this way the dependencies between specifications in a system can be structured. In the Incremental Development Method Package (see Section 6.2.1) this was utilized to plan the increments by specifying which part of the anatomy should be included in each increment. However, the original intention of separating the functional anatomy and the increment planning was not used in the work package method (see Figure 56c). In this method also the specifications within each work package were subdued in favour of a work package dependency structure only. Thus, no separation was made between the dependencies in the system and the work packages by which the system was developed. The reason for this is that it was perceived as too cumbersome to separate these two aspects. A probable consequence of this is that the effort to do re-planning increased. Whether this in fact was the case can however not be concluded from the empiri in this study.
DISCUSSION OF THE FRAMEWORK APPROACH

An example of the use of the Transition Model as a sensitizing guide comes from the unsuccessful project to interface eMatrix with ClearCase through a commercial interface product (see Section 6.6.1 The reference system). During the testing of the interface, severe performance problems were detected. One reason for this was that there was no context separation between the two information systems. Every item in ClearCase was visible in eMatrix and the other way around. From a domain transition point of view this was however not necessary. Only a fraction of the items in ClearCase was of interest in eMatrix and vice versa. If a transitional perspective had been guiding the design of the interface it is likely that the performance problems could have been avoided. A transformation might have been implemented on a transition branch between ClearCase and eMatrix. However, this was never done.

9.1.5 The Stabilizing Core

The Stabilizing Core consists of rules, norms, standards, mores, shared meanings, etc., which provide stability to the coordination domain. In the usage of the Framework the stabilizing core is distributed over all elements. Some examples are

- Rules for identification of products and documents.
- Rules for generating revisions of products and documents.
- Access rights allocated to different user roles (see the example in Section 7.2.2 Actor roles and groups).
- Mandatory elements in Conceptual Model, for example enterprise wide common names on relations, types and policies.

The main purpose of including the Stabilizing Core in the Framework is that it directs the attention to elements which provide a balance between central control and local autonomy in a federated organization of coordination domains. As related in Section 6 The history of the
CHAPTER 9

Framework, three domains were constructed: the S, A and L domains. Each domain was constructed without any common Stabilizing Core. This had the advantage that these domains could be erected quickly and in direct interaction with the development projects using each domain. However, the drawback was that some common core was needed since some projects, for example the UMTS project, impacted several of these domains. Since the federated organization was never erected, such a core was not established.

Still, the need for a balancing stabilizing core on the Ericsson enterprise level is evident. The four major activity domains at Ericsson (see Chapter 2 Research design and realization) make up a federated organization where a balance between central control and local autonomy must be maintained.

9.1.6 The Information System

The evolution of the Information System in a coordination domain is driven by the pragmatic needs of that domain. This is in sharp contrast to many existing IS development methods which suggest a formal specification of the IS before an implementation is done. The intention in the Framework from the outset was that only the motive, outcome and object of the coordination domain (or coordination situations within this domain) should be specified. No overall detailed specification should be needed. The first domain, the S-domain evolved in precisely this way.

However, the experiences show that the evolution has to be controlled in a more stricter way. Frequent changes cause great stress on the users of the IS. Also, since it is very easy to change the implementation in eMatrix there is a risk that the pressure from the user drives the developers to introduce changes which are not analyzed sufficiently. This may result in an application which cannot be maintained, leading to major re-designs. This is what happened at the A-domain.

Another reason for a more stricter control is that the evolutionary development approach clashes with the established, linear development approach at Ericsson. The evolutionary approach is seen as sloppy and unprofessional. As one actor expressed it: “2000 changes in one year is a sign of failure in the requirement specification phase”. In order to counteract these kinds of reactions it is important that a more stringent development method is adhered to without losing the advantages with the evolutionary approach.

In order for the evolutionary approach to work the most important requirement on the Information System is flexibility in the sense that changes, for example adding a new attribute in the Conceptual Model, should be possible to implement in a matter of minutes without halting the database. If such a change takes, say 30 minutes, and the database has to be halted, the evolutionary approach does not work.

The one outstanding requirement on the Information System in the Framework, besides flexibility, is on performance. It must be remembered that the IS is used at globally distributed development sites. Good performance in the vicinity of the server may be disastrous on the other side of the globe. On some occasions, eMatrix was about to be abolished due to poor performance and stability. This was however markedly improved in the later versions
of eMatrix and with the advent of the servlet based web client. However, the vigilance concerning performance must never cease, otherwise there is always a risk that applications are built that degrades the performance anew.

The main conclusion concerning the Information System in the Framework is that it has been demonstrated that an evolutionary development approach of large, globally distributed Product Data Management (PDM) systems is indeed feasible.

9.1.7 The Domain Construction Strategy

The purpose of the Domain Construction Strategy is to provide the actors in the coordination domain with a strategy for how to construct it. The results suggest that the Domain Construction Strategy should be refined into three phases: exploration, trust boosting and expansion. Here is a proposed specification of these phases:

“Exploration”

In this phase the initial construction of the coordination domain is carried out. The main purpose is to achieve shared meaning. One project is the orderer of the work. The construction is carried out by a small (4-7 people) and agile team which includes representatives from the project, the tool vendor and the champion. The champion is an intermediate link between the project and tool vendor who also may be a coordination domain strategist. These poles span different knowledge areas and the interaction between them will provide the thrust in the development:

“I: I’m thinking of this with specification versus constantly changing the implementation. Is this contributing to the positive outcome? R: Yes, well it is the close contact with the users which has made the specification phases very short, you could get direct feedback on it which you then implemented. I: So you have worked more... R: The step more iterative.... the step from ideas for improvements to implementation and then feed-back and the possible adjustments, that cycle has been very fast.” (VDR1-011:15b)

Thus the work is carried out in a ‘daily build’ manner in close interaction among the actors. Requirements are specified on an overall level. Preferably only the outcome of the domain is specified, for example, ‘requirement management shall be implemented’. The team has the authority to decide on any changes on their own without interference from steering boards or reference groups. The work is financed on a risk capital basis. Detailed return on investment analysis is not required since the reliability of such analysis will be low.

“Trust boosting”

The purpose of this phase is to boost the trust about the feasibility of the constructed coordination domain from the exploration phase. Key issues are making actors trust the data in the Information System and to make sure that the performance of the tool is acceptable. This is done in one sharp project, that is, a project which develops a commercial product. All user

2. This is in fact the role I had at the S-domain. The same role was upheld by MT3 and MT4 at the A-domain.
roles around the project are involved and immediate personalized support is provided. The champion plays a key role in this phase. The evolution of the coordination domain is done by controlled changes and consist of fine tuning steps. No major reconstruction of the domain is done. Reference groups and steering boards are consulted and the financing is done on a project basis.

The exploration and trust boosting phases may together be called “establishment” since the overall purpose is to firmly establish an operative coordination domain.

“Expansion”
In this phase several projects are included in the coordination domain. As in the trust boosting phase, the evolution is done by controlled changes. The financing is done by the line organization rather than on a project basis to keep the coordination domain intact between projects.

The three phases suggested above are valid for each coordination situation. If several coordination situations are included in the coordination domain as in the S-domain (see Figure 52, p. 188), it is suggested that the backbone of the domain in terms of coordination situations and their relationships between the situations are constructed to begin with. Each situation may then be constructed more or less independent of the other. Moreover, if a new situation is added to an already existing domain, this can be done in a similar way.

If several coordination domains are established in the manner described, common elements may be extracted to a federate stabilizing core. This in turn may influence the future evolution of the domains. Thus a dialectical interaction between the coordination domains and the core is aimed at. However, it is important that the close interaction between the elements in the Framework is not broken. If for example the Conceptual Model is evolving in isolation from its implementation in the Information System the strategy sketched will not work:

“The last year I think in principle was wasted I must confess because of several things. Really, I think that the proper things weren't attended to. Too much time was spent in... just collecting requirements and specifying things and sort of just in theory, without sitting down and prototyping things and doing things practically. Then you will get this real feedback sort of all the time and you can go near the users and really have something to work with and... it is just that we never got anything to work with, that's the absurdity [...]. I think we are too little focused on the user needs, it was an awful lot of specifying on high levels and requirements which should be collected and that takes an awful lot of time without producing anything that is really useful for the user. So I think that we entirely missed that part”. (VDR1-097:41)

3. This is in fact what happened at Ericsson when the intended reference domain was turned into the central C-domain.
In Figure 57 the deployment strategy is summarized. So far this is only a proposal which has not been verified in practice. However, the findings from the existing domains indicate that this strategy may be a way forward.

9.2 Coordination of development projects

In this section we discuss some impacts on coordination which can be drawn from the Framework intervention in the Ericsson practice.

9.2.1 Impacts on coordination

The consequences described in Chapter 7 Framework consequences show that Framework approach to coordination proposed in this study has had profound impacts on the development practice at Ericsson. To recapitulate, these impacts were accomplished in the following way:

- The coordination definition as proposed by Malone & Crowston (1994, p. 90) was used as a starting point. However, this definition (‘the management of the dependencies between activities’) was considered insufficient for providing operational guidelines for how to coordinate the development of systems of the complexity which can be found in the telecommunication field. To this end we proposed an elaborated conception of coordination as an activity domain (Chapter 4).
- The activity domain is based on the Activity Domain Theory (Chapter 3). In this theory a set of coordination constituents is defined, which elicit fundamental coordination aspects of socially organised human activity.
- The Activity Domain Theory has informed the design of the Framework, where the three Models, the Information System, the Stabilizing Core and the Domain Construction Strategy are derived from the coordination constituents. One of the models, the Process Model, complies with the with the definition of Malone & Crowston. However, we claim that all the elements in the Framework are needed in order to coordinate...
The coordination domain is constructed by the actors in that domain in an experiential learning process where shared meaning as well as coordination models and information system support emerges. Thus both social and technological aspects are considered.

In Chapter 4 *Articulating Coordination* we identified the following needs concerning coordination:

- Interaction and interdependencies must be considered – This is achieved by eliciting the interactions in the Framework models and implementing them in eMatrix.
- Coordination cannot be confined to a rational planning and controlling process only. The social interplay between actors leading to conceptualization, shared meanings, norms and mores must be considered as well – This is achieved through the Domain Construction Strategy in the Framework which promotes the emergence of shared meanings and a stabilizing core in the coordination domain.
- The gap between the design and use of coordination mechanisms must be narrowed – This is achieved by the experiential way of working in the Domain Construction Strategy, where the elements of the Framework are apprehended alternately in the usage domain (the development project which is using coordination) and in the coordination domain (which provide coordination to the usage domain). By a ceaseless focal change between using and designing these elements, the distinction between these two modes is blurred.
- Coordination should be distributed and not confined to a top-down approach – This is achieved by eliciting the separation of contexts in the models and their implementation in eMatrix. For example, in the A-domain the coordination of work packages could be done on the proper level instead of going through the total project manager.
- Coordination processes and structures must be identified – Again, this is achieved by the models and their implementation in eMatrix.
- The coordination mechanisms shall be operational, that is, they should provide methods and tools to meet practical needs – This is evident in the Framework approach.

The pivotal insight in this chain is that all elements in the Framework are needed. If for example the Domain Construction Strategy is not utilized, the construction of shared meaning will be incomplete since the shared meanings are not verified in practice. Furthermore, this strategy presumes that the Information System is easily adapted in concert with the evolution of shared meaning.

### 9.2.2 Balancing central control and local autonomy

‘Balance’ refers to the balance between what should be centrally controlled by corporate units and what should be left to the local autonomy of decentralized organizational units. A key issue in the coordination of complex systems’ development is to find a proper balance between the extremes of total central centralism and total decentralism. A conjecture is that this balance is crucial for several reasons:
DISCUSSION OF THE FRAMEWORK APPROACH

- Total centralism will enable a complete control of the organization. It promotes unified processes, unified information systems and tools, unified shared meanings, etc. Thus total centralism has several appealing features. However, the effort of reaching this state may be prohibitive. Also, the more centralised the organization becomes, the less its ability to react to changes becomes.

- Total decentralism will promote innovations adapted to local circumstances. Furthermore, the ability to react to changes is high. However, the coordination of activities becomes impossible. There is no unification of processes, information systems or meaning.

The results from the study show that local autonomy promotes fast evolution of the construction domain during the exploration phase in the Domain Construction Strategy. This was the case for the decentralized S, A and L domains. However, besides the advantage of quickly achieving an operational coordination support, some drawbacks are also evident. No interaction between S, A and L domains were possible in the sense that there were no common elements in the domains. The Conceptual Models were different, the Process Models were different, etc.

The four coordination domains at Ericsson remained isolated during the course of this study. The intention to unite the existing domains in a federated architecture via the reference system was not pursued, and the course is now set for a single, centrally controlled coordination domain. No well-grounded analysis has been made why the federated architecture was not implemented. Some reasons may be:

- The problems of achieving a shared meaning concerning a federated architecture were too difficult.
- The cost reduction program during 2001 and 2002 inevitably called forth central solutions and one central domain was believed to be cheaper than several domains in a federated structure.
- The importance of shared meaning was by and large neglected which favoured central solutions.
- The group of actors embracing the intentions of the Framework lost influence during 2001 and 2002. The alleged advantages with a federated architecture were not understood by the successors.
- The central solution may in fact be cheaper and more efficient that the federated one.

Thus, the balance tipped from highly decentralized to a highly centralized one where the experiential way of working was abandoned. The consequences of this shift cannot be decisively concluded from the study. However, there are some indications that the lead-times and costs of developing new functionality increased. Other indications of a too far driven centralism would be arduous discussion about meanings, slow reactions to demands for changes, longer times to discover flaws and errors, more pressure of application developers, etc. However, it was not possible to collect the empiri to ground these conjectures within the scope of this study.
CHAPTER 9

9.2.3 Development of large, global information systems

The Framework initiative means that large, global information systems have been developed in an evolutionary way at the S, A and L sites. This development has to a large extent been done in parallel with the actual usage of the information system to support ongoing development of extraordinary complex systems in turbulent environments. The IS development was driven by the pragmatic needs of the projects.

An information system of the size and complexity eMatrix is usually implemented over a number of years in enterprise wide corporate initiatives supported by top management. A conclusion from the Framework initiative is that there is an alternative approach based on the construction of a number of coordination domains in close interaction with actual project needs. This strategy presumes that the IS is exceptionally easy to modify. There are strong indications that this can be achieved with less effort as compared to the traditional linear development method used in similar IS development projects at Ericsson. For example, this is indicated in the following quotation:

“R: I know of no projects which we didn’t run and where the customer had the responsibility, that run as well as this one, really. It has been a successful implementation in any case. I: Can you think of any factors which explains this? R: Well, I think it was so small in scale in the beginning. We focused very hard on doing simple solutions to problems one had. We were very much in focus, we were forced to do solutions which were,... which didn’t require that much work since our resources were so scarce. We found such solutions and I think eMatrix was a prerequisite to create these solutions with these few resources.” (VDR1-009:26a)

In Iivari & Lyytinen (1998) a number of IS development approaches in Scandinavia are compared. These approaches are characterised as ‘grass root’ approaches which emphasise IS evolution and user participation. The research approaches are dominantly anti-positivistic and action oriented. Thus, these approaches are well in line with the approach in this study. However, most of the approaches have had at most a modest practical use: “... despite the practical orientation of many Scandinavian approaches, either because of their roots or action research-based development, most of them are mainly academic exercises.” (Iivari & Lyytinen, 1998, p. 166).

The approach suggested in this study is indeed capable of developing IS which are used in extraordinary complex, practical development situations. This is a feature which clearly distinguishes the Framework approach from the approaches analyzed in Iivari & Lyytinen.

On the other hand, the Framework is geared towards coordination. The paper by Iivari & Lyytinen does not specifically include this criterion which means the applicability of the Framework approach in the areas mentioned in the paper (ibid., p. 162-163) cannot be assessed without further research.

9.2.4 One information system supporting coordination

The conception of coordination as a coordination domain makes it possible to define that domain in such a way that only coordination aspects are salient. In this domain all items are treated as coordination items which means, for example, that the difference between soft-
DISCUSSION OF THE FRAMEWORK APPROACH

ware and hardware, which in other context may be vital, is not noticeable in the coordination domain. Only properties which are important to coordination stand out in this domain.

A consequence of this is that the entire coordination domain can be supported by one and the same IS provided that this IS is capable of implementing the models in the Framework. This has the great advantage that any dependencies between coordination items, for example between requirements and test cases, are managed in one IS only. Usually this is not the case in industrial settings. Requirements may be managed in one IS and test cases in another.

In eMatrix any type of coordination item can be defined. This was most extensively utilized in the S-domain but also to a large extent in the other domains. However, other ISs managing coordination items had taken on the character of ‘black boxes’ according to the Actor Network Theory. This meant that they could not be replaced without generating severe conflicts and thus, the potential of using only eMatrix for coordination support could not be pursued. The results also indicate that this potential was not understood in the Ericsson practice. The conclusion that can be drawn from these observations is that the Framework provides a possibility of greatly enhancing the coordination support. However, this potential was not realized at Ericsson.

9.2.5 Construction of shared meaning

A key element in this Domain Coordination Strategy in the Framework is the establishment of shared meaning concerning the coordination domain in which the IS will be used. Shared meaning is achieved in a small team where users and IS developers in interaction construct the coordination domain by constantly iterating between reflection and action phases. The results indicate that this is in fact what happens when the strategy is applied, for example in the following conversation:

“R: It is when you start communicating that you get a more unified communication, the same way of seeing things and the same measurements of a common tool. I: Well, I think that is an important part, that it mediates some kind of common language in a more clearer way than what has normally been done... R: Yes, and it also has the great advantage this allocation comes very early. I: Then you think about the allocation from requirements to work packages... R: Exactly, you understand who is doing what and who is impacted of which requirements and that is very important. I:...and that you can follow it through to test, etc.? R: Precisely, you have the whole chain. The only thing that we of course wanted was that the tool could produce those overheads as well [laughter]. I: That will be the next step... R: Yeah, we missed that a bit. “(PM5-04-2:33)

9.2.6 Management support - does not have to be top management

The Framework initiative as a whole has not been supported by corporate IS/IT. On the contrary, in 1998, eMatrix was on the brink of being thrown out of Ericsson. The first implementation at the S-domain was initiated shortly thereafter by this author in alliance with a project manager (PM2) for an important project. Now (late 2002) eMatrix is firmly established in the organization. A corporate agreement has been signed with the vendor of eMatrix, Matrix-One and the further development of eMatrix is a corporate concern. One conclusion that can be drawn from these observations is that management support is neces-
sary. However, it does not have to be corporate top management. It is sufficient that an influential project manager supports the initiative. Another conclusion is that a persistent actor can achieve persistent changes in large organizations like Ericsson by firmly pursuing a conviction.

9.3 The Activity Domain Theory

The intention with the Activity Domain theory was to find a suitable theoretical perspective which could be used for analytical and constructive purposes in relation to the main research knowledge interest of this thesis, the coordination of complex systems’ development. A prerequisite for any candidate was that individual / subjective, social and technological / objective aspects should be included.

The activity domain construct is the basis for the design of Framework. For example, the contextuality and orientation constituents are appropriated in the Framework as the Conceptual Model, the tool constituent as the Information System and so on. Thus, the Framework is the link which joins the theoretical and practical perspective and makes the theoretical perspective operational.

The Activity Domain theory can be criticized on several points. First, why exactly these constituents? Surely there must be other coordination constituents which may be considered equally important? I acknowledge this. The set of coordination constituents should be regarded as a proposal whose ultimate relevance will be estimated from their impacts on the usefulness of the Framework in practical settings.

Secondly, these constituents are but too well known and extensively treated in the literature. This is also true. However, as stated above, I have not found any reference in the literature which have combined them into something equivalent of the activity domain. This I claim is a new contribution.

Moreover, in the literature there are several theoretical perspectives reported which have been used to inform the development of IS (see for example Iivari & Lyytinen, 1998). Why not use one of these? One reason is that they appear to lead to only modest practical use (ibid.). Furthermore there seem to be no theory that embraces all of the coordination constituents above (see Section 3.5 Alternative perspectives on human activity, p. 85). Finally, I have worked with the theoretical perspective of praxis as a background to my work at Ericsson which resulted in the Framework and I wanted to sharpen the theoretical background of the Framework.

A conclusion is that the Activity Domain theory has indeed been capable of informing the construction of coordination domains in very complex development situations in the Ericsson development practice.

9.4 Generalizing the results

In this section we will discuss how the results may be generalized. In Section 2.1.6 Generalizations from interpretative research (p. 33) different ways to generalize findings from case studies are discussed.
DISCUSSION OF THE FRAMEWORK APPROACH

General considerations

The Framework as a whole has not been applied in any other company although the IS in the Framework (eMatrix) of course has been used in other companies. Thus, any indications regarding the generalizability of the results are not grounded in any empirical material. A general observation is that the Framework should be applicable in industrial settings similar to the Ericsson one, that is, the industry

- develops complex systems or services,
- runs large, globally distributed projects,
- requires the coordination of many different activity domains,
- and operates on a turbulent market.

This concerns of course the competitors to Ericsson such as Nokia, Siemens, Motorola, etc. It should also be applicable in the manufacturing industries such as the car industry, the air frame industry, etc., as for example indicated by the vendor consultant VDR1:

“There are several things that distinguishes Ericsson from the way we normally try to work in. The implementation from the beginning, then it was rather unique I would say. It was very project centred the model we built, it built very much on managing projects, managing things in the project […] This is not so unique any more… now we have customers working in this way also but then it was rather; then Ericsson was the only eMatrix user in the Nordic countries any way that worked in this way” (VDR1-006:36)

Thus, the Framework may be possible to use in other settings as well. This indication is also strengthened by the fact that the Activity Domain Theory, which the Framework is grounded in, is not specific for Ericsson. This means that we have reasons to believe that the Framework is generalizable on the ‘type’ level. By this we mean the following: The Framework with its Models, the Information System, the Stabilizing Core and the Domain Construction Strategy can be used in other settings as a point of departure for constructing coordination domains in that setting.

Of course, the particular instances of the Framework are unique for each activity domain. This means, for example, that each instance of the Conceptual Model is unique. However, the highest level of the Conceptual Model (see Table 34, p. 217) concerns coordination items such as requirements, products, documents, etc. These items are most likely salient in all development organizations. Thus, it should be possible to transfer this level to other organizations.

Regarding the types of generalizations the Walsham (1995) suggests we can identify the following:

Development of new concepts

We claim that the concepts ‘Activity Domain’, ‘coordination domain’, ‘reinforcement rods’ and ‘Domain Construction Strategy’ are concepts which are either new or used in a new context.

---

4. This IS does not have to be eMatrix. However, an alternative IS must have the same properties as eMatrix, above all the ease of making changes.
CHAPTER 9

**Generation of theory**
The Activity Domain Theory is an example of generation of theory in which new concepts may be a part of a broad network or integrating clustering of concepts, propositions and world-views.

As the results show, this theory is applicable within the Ericsson development practice. Thus, the theory has acquired the status of a ‘bridge-head’ within a specific domain of application (Alvesson & Sköldberg, 1994, p. 32). The generalization of the theory is validated by enlarging the application areas successively within the domain. In Section 10.7.6 *Product life cycle management – towards a new foundation* (p. 239) one such enlargement is suggested.

**Drawing of specific implications from in-depth case studies**
Some implications that can be drawn from the study are:

- The domain construction strategy provides a mechanism to develop agile PDM systems which can be adapted to changing circumstances.
- The entire coordination domain can be supported by one information system.
- The Domain Construction Strategy provides a mechanism to include the generation of shared meaning in the construction of the coordination domains.

**Contribution of rich insight,**
As stated in Section 2.2.3 *Data triangulation* (p. 40) the empirical material in the study is extensive: nearly 500 interview statements, 270 documents, 1000 e-mails, notebooks, etc. It is surely possible to read this material in several ways, thus giving rise to new insights.

As a final reflection on generalizability I have deliberately refrained from value statements regarding the Framework and concentrated on describing and analysing the consequences of the Framework intervention in the Ericsson practice. In the final reading it is very much up my readers to judge if the results are applicable in other situations as well.
10 Conclusions

In this chapter I will discuss the conclusions that can be drawn from the study. What is the essence of the findings? What is the significance of the study for academy and industry? How can it be placed in a broader context? Which are its limitations? What areas for future research does it open up?

10.1 Recapitulation of research questions

The purpose and the main research question of this study were formulated as follows:

> When developing complex systems subject to changing presumptions, the coordination of the development tasks is a crucial activity. A Framework has been developed and deployed in the development organization at Ericsson with the intention of supporting the coordination. The purpose of this study is to develop knowledge of the impacts of the Framework on the coordination task. The main research question is: What are the impacts on coordination from the Framework?

In Table 35 the detailed research questions and corresponding knowledge contributions are recapitulated.

<table>
<thead>
<tr>
<th>Detailed research question</th>
<th>Knowledge contribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>RQ2: What are the overall consequences from the Framework intervention in the Ericsson development practice?</td>
<td>Identified consequences</td>
</tr>
<tr>
<td>RQ3: Which elements in the Ericsson development practice contribute to successful outcomes of development projects?</td>
<td>Identified elements</td>
</tr>
<tr>
<td>RQ4: What are the impacts on coordination from the Framework intervention in the Ericsson development practice?</td>
<td>Identified impacts on coordination</td>
</tr>
</tbody>
</table>

10.2 Conclusions concerning the impacts on coordination

In this section the conclusions concerning the impacts on coordination from the Framework intervention are given.

10.2.1 Profound impacts on coordination from the Framework intervention

The answer to the main research question is given in Section 9.2 Coordination of development projects together with the discussions concerning the generalizability of the results in Section 9.4 Generalizing the results. The approach to coordination proposed in this study has had profound impacts on the development practice at Ericsson. As described in Chapter 6 The history of the Framework, in less than three years from may 1999 until mid 2002 the number of projects impacted rose from zero to around 140, distributed over more than 20

---

1. Some of these results have been reported in Taxén (2002).
development sites worldwide. The profoundness of the impacts is captured in the following quotation:

“R: Especially for the execution part I think we would not have been able to run this project without the tool. I think if you simply look at the number of work packages, the number of products that we have delivered, the number of deliveries that we have had, if we would have had to maintain that manually, that would have been a sheer disaster. [...] we had some, only in my part of the project, some 200 work packages or work packages groups or whatever you want to call them, deliveries, on the average 2-5 subprojects within them 5-10 blocks being delivered, just keeping track of that [...] would have been a hell of a job.” (PM4-05-1:07).

10.2.2 Balancing central control and local autonomy - a key issue

‘Balance’ refers to the balance between what should be centrally controlled by corporate units and what should be left to the local autonomy of decentralized organizational units. A key issue in the coordination of complex systems’ development is to find a proper balance between the extremes of total central centralism and total decentralism. The balance between centralism and decentralism is controlled by a stabilizing core, the extent of which can be modified to achieve an optimum balance. As long as this control mechanism is not in place, the organization is destined to oscillate between the equally destructive alternatives of total centralism and total decentralism.

As intended, the Domain Construction Strategy promoted a fast evolution of the decentralized S, A and L domains. However, the price was that no interaction between S, A and L domains was achieved. The reference system, which was intended to be the stabilizing hub in the federated architecture of coordination domains, was transformed into the single, centrally controlled C-domain.

The conclusion of the study is that the original intentions in the Framework concerning a federated coordination domain architecture were not implemented. Instead, the balance shifted from total decentralization all the way to total centralization. However, it remains to be investigated if this shift indeed provides the expected benefits in terms of efficiency and cost savings. On the contrary, recent research results indicate that organizations which are able to maintain a proper balance will be strongly armed to manage the inevitable change process (Davenport et al., 1992; Sawhney & Prandelli, 2001; Sage, 2001).

10.2.3 Evolutionary development of global information systems - a reality

Information systems of the size and complexity of eMatrix and the Corporate information system C-PDM are usually implemented over a number of years in enterprise wide, ponderous corporate initiatives supported by top management. A conclusion from the Framework initiative is that there is an alternative, agile approach which is based on the construction of a number of coordination domains in close interaction with actual project needs. The approach presumes that the information system is extraordinarily easy to modify. However, if this approach shall not result in a number of detached domains, it is imperative that corporate units are in control of the stabilizing core in order to achieve a proper balance between enterprise needs and local autonomy.
10.2.4 Coordination domain construction in three phases

How should a new coordination domain be constructed? How should the models, the stabilizing core, the information system and shared meaning evolve when the Framework is applied in a new setting or in a new coordination situation? A conclusion from this study is that it should be done in three phases:

- **Exploration** – In this phase the main purpose is to achieve shared meaning. The construction is carried out by a small (4-7 people) and agile team which includes representatives from the project, the tool vendor and a coordination domain strategist. The work is done in a daily build manner in close interaction among the actors. Requirements are specified on an overall level.

- **Trust boosting** – In this phase the key issues are making actors trust the data in the Information System and making sure that the performance of the tool is acceptable. This is done in one sharp project, that is, a project which develops a commercial product. All user roles around the project are involved and immediate personalized support is provided. The evolution of the coordination domain is done by controlled changes and consists of fine tuning steps.

- **Expansion** – In this phase several projects are included in the coordination domain. As in the trust boosting phase, the evolution is done by controlled changes. The financing is done by the line organization rather than on a project basis to keep the coordination domain intact between projects.

10.2.5 One information system for coordination provides major advantages

A conclusion of the Framework approach is that all coordination situations (see Section 7.2.1 Coordination situations, p. 155) in a coordination domain can be supported by the same type of information system, which at Ericsson was eMatrix. This was used most extensively at the S-domain. Such an arrangement has several advantages as compared to a situation where different information systems are involved:

- Interfaces are not needed between different information systems in different coordination domains. For example, the coordination of requirements, incremental development and test cases can all be done in the same information system.

- A number of information systems can be replaced by one. For example, different requirement management systems can be replaced by implementing the requirement coordination situation in eMatrix.

- The coordination information will be consistent. Unique information elements will be stored in one information system only.

- Complete traceability of dependencies between coordination items defined in the Conceptual Model is achieved.

- There is a homogeneous world-view for the coordination domain. This alleviates the construction of shared meaning in the elaboration phase of the Domain Construction Strategy.

- Changes which are common to all coordination situations are easily implemented, for example adding an attribute to all coordination items.
• The agility in reacting to imposed changes is higher. The information system applica-
tions can evolve with the evolution of the needs of the coordination domains.
• New coordination situations can be included, for example, to replace legacy informa-
tion systems.
• In each coordination domain a balance between central control and local autonomy
can be upheld by access rights, imposed rules, inheritance mechanisms, etc.
• The establishment of a federated structure of coordination domains is made easier
since all coordination domains share the same basic information system.

This does not imply that there is one information system for all tasks in the organization.
There will still be a need for specialized information systems, for example for software con-
figuration management. Thus, a number of interfaces between the Information System in
the Framework and other information systems are still needed. However, these interfaces
will be placed differently in the information system architecture.

10.2.6 Shared meaning and coordination support evolve simultaneously
The results indicate that the effort of achieving shared meaning is usually not elicited in
development practices although it might well be a substantial, if not the major part of the
overall cost of providing coordination support for development projects. If this cost is not
included in cost calculations, erroneous decisions may be the result. For example, in choos-
ing between a federated information system architecture and a centralistic one, the cost of
achieving shared meaning may be crucial. Thus it is important to elicit this cost and find
strategies to reduce it.

A conclusion from the study is that the Domain Construction Strategy in the Framework
provides such a strategy. In the S, A and L domains the close interaction between actors pro-
moted shared meanings, signifying models, a stabilizing core and information system sup-
port. By alternating between the usage domain and the coordination domain the gap
between detached reflection and practical usage is narrowed. Conjectures can be discussed,
implemented, tested, reflected upon and modified continuously. This approach makes it pos-
sible to conceptualize the coordination domain and firmly ground this conceptualization in
the concrete practice of developing complex systems.

10.3 Conclusions concerning the reinforcement rods
The reinforcements are elements which experienced actors within Ericsson, or in close
cooperation with Ericsson, consider important to the successful outcome of development
projects. The following reinforcement rods were identified:

• **Evolution**: New knowledge acquisition should be done gradually and build on an
  existing knowledge base.
• **Orientation**: Everybody in a project needs to have a common apprehension about what
  the project is all about and where her position in the whole complex is.
• **Federalism**: A development work should be divided into small, more or less independent
  areas in cooperation.
CONCLUSIONS

• **Balance**: The balance between order and disorder must be upheld in the project.
• **Isomorphism**: There is a need to define mappings between different world-views.
• **Shared meaning**: The actors must share some meaning about the context in which they act.
• **Participation**: All participants shall feel that they are participating in the project and that no one is more important than the other.
• **Focus**: All necessary resources should be aligned towards a clear and concrete target: what should be delivered at a certain point in time.
• **Turning points**: This refers to unexpected events which have a decisive influence on the future trajectory of the project.

The conclusion from the study is that the Framework manifests of all these rods except ‘participation’, ‘focus’ and ‘turning points’.

10.4 Conclusions from the history of the Framework

Several conclusions can be drawn from the history of the Framework:

• Entrepreneurs are necessary in order to instigate an innovation in an organization. Examples of such entrepreneurs are the project managers PM2 and PM3 who were promoting the Framework in their projects. As the instigator of the Framework I would also consider myself as an entrepreneur.

• New ideas and innovations have a tendency to flourish for a certain period in the organization, then fall into oblivion and eventually reappear in a different form. The Incremental Development Method Package is an example of this. After being used in several projects during 1996-1997 it was obliterated in early 1998. During 1999 and 2000 basically the same method reappeared under the name ‘work package method’. Another example is the Specification Based Data Model (Gandhi & Robertson, 1992) which was used in the AXE-S project. After the termination of that project, this model reappeared later as the anatomy construct.

• The trajectory of an innovation in an organization is highly unpredictable. As the analysis of the history shows, the different elements of the Framework evolved as detached element during the period before 1997. Between 1997 and 2001, the elements were joined as systemic elements in the Framework, that is, they were all interacting and contributing to the overall properties of the Framework. After 2001 the elements once again retracted to detached, individual elements. Thus, the only period when the Framework as a systemic whole existed at Ericsson was between 1997 and 2001.

10.5 Additional conclusions

*The Activity Domain Theory*

The Activity Domain Theory has proven capable of informing the design of the Framework which in turn has shown to be operational in the Ericsson development practice. Since an information system is included in the Framework, the Activity Domain Theory is also capa-
ble of informing the development of information systems where both social and technological aspects are included.

**An operative conception of coordination**

The results show that it is feasible to conceive coordination as a particular activity domain which provide coordination support for development projects. This enables coordination to be experientially constructed by the actors in this domain, which means that individual knowledge, shared meaning and objectified organizational artefacts such as models and information system support evolve in a dialectical manner.

**Applying the Framework in the research work**

The Framework has been applied to itself in the research concerning the Framework. The conclusion from this application is that the recording and analysis of a large empirical material are feasible. This opens up the possibility for the further usage of eMatrix as a research tool for supporting the analysis of the data in interpretative research projects.

### 10.6 Research design and method evaluation

In this section I will evaluate the research design and methods used for the different research questions.

**RQ1: How did the Framework evolve in the Ericsson development practice?**

This research question is treated in Chapter 6 *The history of the Framework*. The method used was interviews and document analysis. The documents consist of internal Ericsson documents, e-mail, personal notes, etc. No restrictions were imposed on the material besides that classified documents, financial figures and explicit lead-times of development projects could not be used. Since I instigated the Framework and actively have pursued its deployment I have had a first hand access to the data, both as an observer and as a promoter of the Framework.

However, the close access to the data has a severe drawback: the bias in the selection and interpretations of the material. I have tried to counteract this by accounting for both positive and negative events. I have also used an abundance of quotations to illustrate my points. I have also strived to ground these in other data sources wherever possible. Another drawback is that I have not been able to validate my findings with some actors due to the turbulent situation at Ericsson during 2002.

**RQ2: What are the overall consequences from the Framework intervention in the Ericsson development practice?**

This research question is treated in Chapter 7 *Framework consequences*. The method used was a modified version of the Grounded Theory (Strauss & Corbin, 1998). The coordination constituents of the Activity Domain Theory as well as the Framework elements were used as a guide in the analysis.

I have used the Framework in the construction of a ‘coordination domain’ for my own research. This means that I defined a Conceptual Model (see Figure 11, p. 46) and imple-
CONCLUSIONS

mented this in eMatrix. The definition followed the strategy in the Framework: evolution by a constant iteration between the model and its implementation. In fact, around 15 revisions of the Conceptual Model were defined during the analysis. Any generated category is easily traced to its grounding data, as well as my interpretation of the grounding. Thus the way the theory is generated is in principle controllable.

Without this analysis support it would have been tedious and hard to keep track of all the dependencies among the more than 120 categories and 600 data elements used in the analysis. All the tables in Chapter 7 Framework consequences as well as the table in Appendix 1 were generated as reports from eMatrix (with some additional editing to fit the text editor used in the dissertation: FrameMaker).

One drawback of this way of working is that the ease of defining new categories ‘on the fly’ may lead to an abundance of categories, some of which may be superficially or vaguely defined. There are certainly traces of this in the Appendix 1 and the analysis can no doubt be improved. However, I claim that the essential consequences from the intervention of the Framework have been identified.

RQ3: Which elements in the Ericsson development practice contribute to successful outcomes of development projects?

This research question is treated in Chapter 8 Reinforcement rods. Again, the method used was a modified version of the Grounded Theory. I mean that this has worked well. The results have been confirmed by PM1, one of the interviewees. Some weaknesses are that the number of interviewees is fairly small and that the Activity Domain Theory may have influenced my identification of the reinforcement rods in an unreflecting way. Certainly, elements in this theory were part of my preconceptions. However, as can be seen from the interview questions in Section 2.2 Realization, these elements were not particularly salient.

RQ4: What are the impacts on coordination from the Framework intervention in the Ericsson development practice?

The knowledge contribution from this research question is derived from the results in Chapter 6 The history of the Framework (RQ1), Chapter 7 Framework consequences (RQ2) and Chapter 8 Reinforcement rods (RQ3). These results are discussed in Chapter 9 Discussion of the Framework approach which thus provides an answer to RQ4.

The method used may again be characterized as a modified version of the Grounded Theory. However, in this case the theory preconception is outspoken since the coordination constitutions of the Activity Domain Theory as well as the Framework elements were used as a ‘search grid’. No additional data was used. Instead, the knowledge contribution from RQ4 should be regarded as penetrating the results from the previous research questions from the coordination point of view. Thus, the same method strengths and weaknesses found in the methods used for the previous research questions apply for RQ4 as well.

However, since the answer to RQ4 provides the central knowledge contribution of this study some additional dangers lurk in the shadow of this research question. One is the risk of over-

---

2. In practice, a controller needs to have access to a eMatrix installation.
modesty or self-aggrandizement concerning my own role. I have tried to meet this by clearly stating my contributions to the results.

10.7 Further areas of research
In this section, I will discuss a number of possible future research areas.

10.7.1 Investigating the application domain of the Framework
The Framework has been applied only in the Ericsson development practice. However, the Activity Domain Theory applies in principle to any human activity that needs to coordinate its work. A topic for future research would be to identify the application domain of the Framework approach, possibly by applying it in other settings than Ericsson.

10.7.2 Evolutionary information system development
The IS development approach in the Framework can be characterized as evolutionary and pragmatical. No detailed specification exists. Instead the IS unfolds in concert with other elements the coordination domain such as the Conceptual Model, the Process Model and the evolution of shared meaning. An observation from the Framework use at Ericsson is that this way of developing IS is not without problems. More research is needed in order to define a development process by which evolutionary and pragmatic development of IS can be carried out. This has also been noted by Kristensen (2001).

10.7.3 Consolidate reinforcements
The knowledge contribution concerning the reinforcement rods was generated from the Ericsson practice. A natural topic for future research is to further anchor these results in the literature and in other settings beside Ericsson.

10.7.4 The optimum balance between control and autonomy
The results indicate that the balance between control and autonomy may be critical for an organizations' ability to adapt to changes. A line of research would be to investigate the nature of that balance and how an optimal balance can be achieved. Since population level theories are one branch of theories concerning organizational learning (Lewitt & March, 1988), one starting point might be the natural evolution based stochastic optimization algorithm called Gaussian Adaptation discussed in Section 3.3.7 Stabilizing core (Kjellström & Taxén, 1981).
10.7.5 A systems development method based on the Framework

The system development method suggested in Section 6.1.2 *The system design environments*, p. 121, was never implemented at Ericsson. The key element in this method is to separate development contexts by the specification – implementation construct of the Specification Based Data Model (Gandhi & Robertson, 1992). In heterogeneous systems these contexts may implement a specification in different technologies, for example software, hardware or intermediate forms like programmable hardware.

Each specification would be implemented by the aid of process components which are aligned with the technology used in the context (see Section 6.1.2 *The system design environments*, p. 121). For example, a software implementation might employ a particular part of the Rational Unified Process, while the placement and routing of an ASIC would employ a CAD-system.

Thus, rather than adhering to well-known development strategies such as 'top-down', 'bottom-up' or 'meet-in-the-middle' ones, the development would be driven by the status of the specifications and implementations. Development work may in principle be going on at any level in the system simultaneously (Taxén 1995).

10.7.6 Product life cycle management - towards a new foundation

A project was started during 2002 to provide a new foundation for the product life cycle management (PLM) at Ericsson (Taxén & Svensson, 2003). This project took as its base the Activity Domain Theory. The coordination constituents in this theory and its corresponding operational elements in the Framework were accepted as the basis for grounding the plethora of concepts regarding PLM which exist in the organization. Furthermore, an analysis using the Activity Domain Theory as a ‘sensitizing device’ revealed several inconsistencies in the existing foundation. These had mainly to do with an omission of interaction between the coordination constituents. However, the results so far show that the Activity Domain Theory is a useful analytical tool with respect to PLM at Ericsson. Thus, another future research area would be to determine the application domain of the Activity Domain Theory.

10.7.7 Organizational learning

In spite of many years of research it seems that practical guidelines for achieving organizational learning are rare (Huber, 1991; Virkkunen & Kuutti, 2000). The root cause of this appears to be the vagueness of the ‘organization’ concept. To this end Virkkunen & Kuutti (ibid.) suggest an alternative unit of analysis, the Activity System as defined in the Activity Theory (Engeström, 1999).
CHAPTER 10

Since the Activity System bears many similarities to the construct of activity domains as defined in this study, a topic for future research would be to investigate the relation between activity domains and organizational learning.

The suggested line of research is as follows: The concept of organizational learning is closely related to organizational knowledge. In order to define organizational knowledge some criteria for what constitutes this knowledge are needed. The epistemological and ontological positions taken in this study imply that only individuals can act in the organization. A statement like “Ericsson designs a new mobile telephone” is a sign of reification of the organization, that is, it is attributed with acting powers of its own.

This means that organizational knowledge is associated with individuals in a particular organization. However, not all individual knowledge can be called organizational. For example, knowledge of how to develop object-oriented software might be necessary for the Ericsson organization to fulfill its goals. However, this type of knowledge cannot in our opinion be considered organizational knowledge, since it can be utilized in many organizations. The bottom line must be that organizational knowledge is only effective in conjunction with a particular organization. I suggest that organizational knowledge must possess at least the following characteristics:

• It must be tightly related to the purpose of the organization and its goals.
• The knowledge concerns all the actors in the organization to some extent.
• It must be objectified as descriptions, rules, norms, institutions and the like which are valid for the entire organization. Otherwise, it is not possible to maintain the knowledge between generations of actors.

Knowledge concerning how to coordinate the development of complex systems fulfills these criteria. This knowledge is of vital importance for the organization since it directly concerns the purpose of the organization of delivering products or services to a market. Thus, the learning that takes place in the construction of a coordination domain may be considered as one type of organizational learning. Some preliminary results have been reported in Taxén (1999).

10.8 Concluding remarks

In this section I will discuss the quality criteria of the dissertation, give a personal account of my life as an industrial Ph.D. student and finally, summarize the knowledge contribution of the study.

10.8.1 The quality criteria of the dissertation

In Chapter 2 Research design and realization I stated some quality criteria which a dissertation should meet:

• Its knowledge contribution should be original.
• The findings should be credible and controllable.
• The dissertation should be communicable.
• The findings should be relevant.
CONCLUSIONS

Originality
First of all, I claim that the Framework is original. Its elements taken one by one are certainly not original, but to the best of my knowledge they have not been conflated in the particular constellation they have in the Framework. For certainty, this is valid for the Ericsson practice.

The same goes for the Activity Domain Theory. The coordination constituents are well known one by one, but again, not brought together into the constellation of Activity Domains.

However, I claim that the essential originality lies in the combination of the Activity Domain Theory and the Framework. As this study demonstrates, this combination has resulted in concrete artefacts, including global information systems, which have been used on the ‘battlefield’ with promising results.

The knowledge contribution from research question RQ1 (the history) is certainly original since it concerns a particular course of events in a particular setting. This is valid also for the knowledge contribution from research question RQ2 (the consequences). However, some of the findings from RQ2 are certainly known previously, for example, the importance of performance of the information system in order to get users to accept the system.

Concerning the knowledge contribution from research question RQ3 (reinforcements) I do not claim that the reinforcement rods are original as such. The purpose of this research question was to identify elements in the Ericsson development practice, which according to experienced actors contribute to successful outcomes of development projects. From the discussion in Section 8.2 Selective coding - reinforcements related to the Framework it is clear that the Framework is capable of embedding some of the reinforcements in development projects.

Finally, I claim that the knowledge contribution from the research question RQ4 (impacts on coordination) is original since this contribution is directly related to the Framework.

Credibility and controllability
I have strived to assure credibility in various ways. First of all, I have used an abundance of quotations and references to internal Ericsson sources. I have tried to select interviewees in such a way that I cover as many aspects as possible. Both advantages and disadvantages have been accounted for. Furthermore, I have openly declared my own perspective on research positions taken, for example concerning ontology and epistemology. I have also accounted for my bias concerning the Framework.

Controllability has been assured by the traceability in the eMatrix research database used for the analysis. Every category can be traced to its grounding data as well as my interpretation of the grounding. The internal Ericsson material is not freely available. However, it is accessible on request from Ericsson.

Communicability
Writing a communicable dissertation is certainly a challenge since the author is dependent on the preconceptions of the reader. I have tried to write in a concise and clear style. In the
introduction (see Section 1.9.1 Reading guidelines) I have indicated what parts different stakeholders might be interested in. This should give some idea about the preconceptions needed. Some parts are easier to access, for example Chapter 6 The history of the Framework. On the other hand, Chapter 3 The Activity Domain Theory is quite challenging.

I have tried to organize the study in a straightforward way to enhance communicability. I have also occasionally repeated the same piece of information to improve the overview. I have also used a fair amount of figures and examples which also should contribute to the communicability. Finally, I have tried to avoid internal Ericsson jargon as much as possible.

Relevance
It should be obvious from the study that the results are relevant for Ericsson. The results should also be relevant to other, similar organizations. However, it outside the scope of this study to investigate this.

10.8.2 Limitations of the study
The main limitations of the study are:
• The data is collected from one study field only. Thus any generalization of the results is at this moment a conjecture only.
• The interpretations of the data made in the study may be biased, incomplete or inadequate. However, the interpretations done can be accounted for and traced to its grounding data in the research database.
• The evolutionary information system development approach is limited to large, global IS for coordination purposes in the study. Whether these results are transferable to other types of ISs cannot be concluded from the study.

10.8.3 Significance for Ericsson and other industries
The impacts from the Framework on coordination in the Ericsson practice have been profound as discussed earlier. However, the intentions of the Framework were not pursued. The close interaction in the Framework strategy between the models, their implementation in the IS and the immediate feedback from the development practice was abandoned in favour of a centralistic approach, more detached from practice.

This meant that the potential of continuously improving the coordination of development projects was curbed. One example is the potential of eschewing progress reports in favour of a continuous monitoring of the status of the project:

“R: Some project manager gave me the example of a subprojects manager that said: “Yeah but you don’t need a progress report, you can go into eMatrix, everything is there!” (MT4-10-1-27)

In addition to this, other more deeply going potential improvements were lost such as establishing a federated IS architecture and providing ‘one tool solution’ for coordination domains.
CONCLUSIONS

A conclusion of these observations is that the Framework as a coherent approach to coordination does no longer exist at Ericsson in spite of the seemingly firm establishment of the eMatrix information system. Moreover, this means that the one outstanding distinguishing quality of eMatrix, its flexibility, is not utilized to its full potential at Ericsson.

As stated, the applicability of the Framework in other settings has not been investigated in this study. However, it is likely that the Framework can be applied in product development organizations with similar structure as Ericsson.

10.8.4 The industry Ph.D. student - some reflections

I have carried out my studies as a member of the Industry Research School at the Department of Computer and Information Science at Linköping University. This means that I have shared my time between Ericsson and the Research School.

After a long period of work in the industry it is certainly very rewarding to have the possibility of reflecting on the often very goal-oriented industrial work. In the industry there is seldom room for a deeper analysis of the work and to ground decisions firmly. On the other hand, in academia, there is always the problem of getting a good access to real, practical settings where theory and prototypes can be put to harsh tests. To oversimplify, there is a lack of theory in industry and a lack of practice in academia.

These problems are all but too well known and it seems to be difficult to find good ways of cooperation between industry and academia. The most appealing feature with the Industry Research School is that it provides the same person to switch between these areas. On many occasions findings from my research are directly interesting for the industry. One example is the articulation of the reinforcement rods, which is imminent in the Ericsson practice but only in an unreflecting way. Conversely, the long industrial experience I have has been a very useful background in my research, for example, as case studies for course reports.

However, from my perspective there has been an apparent bias in this relationship: the academia seems more interested in the industry than the other way around. I have not been able to interest Ericsson for my research with the exception of a few people. One reason may of course be my own inability to convey the results. Another reason may be that the research was carried out in a setting where the focus is on delivering products in time. There is simply not much leeway for reflections. Still another reason may be the nature of the study field. Coordination including information system support is considered not to be core business at Ericsson. As one employee expressed it: “When you are working with methods and tools you always sit in the sidecar.” This situation is very different from the Corporate Research units at Ericsson where cooperation with academia is natural both for Ericsson and academia. In spite of this, the disinterest from Ericsson is remarkable. At the end of the day, coordination is imperative to get the products out on the market.

It is my firm belief that an arrangement like the Industry Research School is a first step in the right direction to closer cooperation between industry and academia. However, if this institution is to soar, there must be a formalized arena where the research results can be evaluated, scrutinized and eventually internalized in the organization – a place for mutual cooperation, discussion and genuine exchange of experiences.
10.8.5 Summary of the knowledge contributions

In summary, a Framework for coordinating the development of complex systems has been designed and instigated in the Ericsson development practice by this author. The Framework is grounded in a tentative theory called the Activity Domain Theory, which in turn is based on the praxis philosophy. In this theory the interaction between the individual and her environment is mediated by signs. This means that subjective as well as objective aspects are considered, which I claim is a prerequisite for coping with complex development tasks. Coordination is conceived as a particular activity domain which provides coordination to the development projects. The coordination domain is experientially constructed by the actors in this domain, which means that individual knowledge, shared meaning and objectified organizational artefacts such as models and information system support evolve in a dialectical manner.

The data has been collected from the development practice at Ericsson. From this practice a number of reinforcement rods have been identified which contribute to the successful outcome of projects, some of which are supported by the Framework.

The Framework has unfolded in this practice over a period of more than 10 years. It has had profound impacts on the coordination of extraordinary complex system development projects.
11 References

The references from Ericsson are the property of Ericsson. They are accessible on request through this author. Ericsson reserves the right to approve or refuse the access as well as the the terms for accessing them.

11.1 From the literature


Chalmers A F (1976): What is this thing called science?, University of Queensland Press, St Lucia, Queensland.


CHAPTER 11


REFERENCES


REFERENCES


CHAPTER 11


REFERENCES


CHAPTER 11


11.2 Ericsson references
ERN-1989-04-03: Hanteringsprinciper (in Swedish)
ERN-1991-04-03: Principer för informationshantering (in Swedish)
ERN-1991-08-08: Systemelementbeskrivning för DIM (in Swedish)
ERN-1992-12-21: Strategi för processutveckling (in Swedish)
ERN-1993-09-23: Definition of concepts in SDP
ERN-1994-04-26: Om IS-utvecklingen i AXE-S (in Swedish)
ERN-1994-06-02: Ang CM issues (in Swedish)
ERN-1994-08-15b: Bokslut - minnesanteckningar (in Swedish)
ERN-1995-03-15: OH bilder (in Swedish)
ERN-1995-03-18: HWD 1.0 (Hardware Development release 1.0)

252
REFERENCES

ERI-1995-07-06: Arkitektur och principer för IS i AXE-S (in Swedish)
ERI-1996-12-02: Incremental Development for AXE10
ERI-1995-12-12: Mina erfarenheter från SDS-projektet (in Swedish)
ERI-1996-12-02: Incremental Development for AXE 10
ERI-1997-10-20: Positive feedback from ETM
ERI-1997-10-23: Incremental design, CMS 30 ph7, IDMP
ERI-1997-11-05: Great success at EPK with CPLtool
ERI-1997-11-10: System management at UAB orders the use of PDL
ERI-1997-12-17: ID WS summary
ERI-1998-04-20: Experience report from Polaris Charging
ERI-1998-06-26: Report from the Matrix WS at JDI, Montreal
ERI-1999-02-15: IS/IT steering board
ERI-1999-05-31: IT/IS steering board
ERI-1999-10-21: Interface LME/DTL meeting
ERI-1999-12-01: UMTS product dev strategy
ERI-2000-01-11b: Information object management systems
ERI-2000-02-10: Evaluation of ematrix and rational tools
ERI-2000-03-17: EMW comparison eMatrix - C-PDM
ERI-2001-04-20: MARS business case
ERI-2002-07-31: Organization chart
ERI-2002-06-06: eMatrix usage at Ericsson

11.3 Ericsson references (Taxén)
LTX-1994-04-01: Coordination of the SW-HW process.
LTX-1994-07-06: Vidareutveckling av SDP-arkitekturen (in Swedish)
LTX-1995-04-26: SDP Basics
LTX-1996-02-23: A Coherent Framework for Development
LTX-1998-10-12: Requirement specification of a tool for managing ID
LTX-2002-03-28: Common Information Model.

11.4 World wide web
WEB-1: Home page of MatrixOne: http://www.matrixone.com
Appendix 1: Grounding of effect categories

In Table 36 below the grounding of the effect categories is given.

Table 36. The grounding of effect categories

<table>
<thead>
<tr>
<th>Category</th>
<th>Category description</th>
<th>Grounded in</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>especially interesting quotations</td>
<td>VDR1-028:59, PM4-05-1:07, PM2-45:43, PM4-11-1:20c, PM2-10:50b, PM2-10:50c, PM3-36:42, PM3-41:40a, PM3-41:40b, IT2a-97:16a, IT2a-21:43b, MT3-56:52, MT3-35:54, MT3-12:40, MT3-07:03, MT4-13-4:48b, MT4-09-4:57, CM1_2-33:09, CM1_2-24:49, MT1-68:04, MT1-36:30, DE1-05, ERI-2002-06:06, ERI-2001-12-12, ERI-2000-07-06, ERI-2000-06-01, ERI-1999-12-08b, MT4-16:2:05, MT4-16:0:25, MT4-09-0:01a</td>
</tr>
<tr>
<td>activity domain,</td>
<td>issues that are related to activity domains in general</td>
<td>MT4-18-2:21, MT4-16-4:39, MT1-44:08, MT1-23:31</td>
</tr>
<tr>
<td>adaptation,</td>
<td>issued related to adaptation of the tool, processes or models</td>
<td>ERI-2002-08-01:2, ERI-2002-08-01:4, VDR1-071:03, VDR1-073:07, IT2b-35:12, IT2b-33:30, ERI-1995-01-27, PM2-63:26</td>
</tr>
<tr>
<td>advantages &amp; problems,</td>
<td>issues related to any impacts of the FWK that cannot be labelled positive or negative</td>
<td>PM4-06-4:04, VDR1-031:35, CM1_2-20:10b, PM2-61:01, PM3-48:57, PM3-17:45, PM3-62:19a, VDR1-031:35</td>
</tr>
<tr>
<td>advantages,</td>
<td>issues related to positive impacts of the FWK</td>
<td>PM4-05-1:07, PM4-05-2:13, PM4-05-3:33, PM2-45:43, PM4-05-4:52a, PM4-05-4:52b, PM4-06-1:13a, PM4-06-1:13b, PM4-06-2:29a, PM4-06-2:29b, PM4-06-4:04, PM4-07-0:37, PM4-07-4:24, PM4-08-1:15, PM4-08-3:55, VDR1-009:26a, VDR1-009:26b, VDR1-011:15, VDR1-020:41b, VDR1-030:12, VDR1-031:35, VDR1-047:30, VDR1-052:41, VDR1-053:18, VDR1-055:10, VDR1-078:45a, VDR1-078:45b, VDR1-078:45c, VDR1-087:48, VDR1-085:23b, VDR1-083:30a, VDR1-109:23a, MT1-41:06, VDR1-009:26a</td>
</tr>
<tr>
<td>agile teams,</td>
<td>issues related to small and fast teams. Often when the development has been fast there are few actors involved and no steering groups.</td>
<td>MT1-41:06, VDR1-009:26a</td>
</tr>
<tr>
<td>anatomy,</td>
<td>Issues related to anatomies</td>
<td>PM4-07-0:37, RES1-01, ERI-2002-07-01:1, PM4-12-3:43a, PM3-70:13, PM3-09:07, PM3-23:15, PM3-29:34, PM3-41:40b, PM3-41:43b, ERI-1998-01-01, ERI-1997-09-05, ERI-1994-12-01</td>
</tr>
<tr>
<td>articulation,</td>
<td>of things not well thought through. The tool prompts an interpretation.</td>
<td>CM1_2-51:00</td>
</tr>
</tbody>
</table>
## Table 36. The grounding of effect categories

<table>
<thead>
<tr>
<th>Category</th>
<th>Category description</th>
<th>Grounded in</th>
</tr>
</thead>
<tbody>
<tr>
<td>champion.</td>
<td>Issued related to the champion role in the deployment</td>
<td>MT4-07-0:50, CM1_2-33:53</td>
</tr>
<tr>
<td>change.</td>
<td>Issues related to change in the praxis</td>
<td>VDR1-077:24, PM1-32:10a, PM1-35:08a, PM1-37:12, PM3-41:40b</td>
</tr>
<tr>
<td>communication.</td>
<td>Issues regarding communication between actors.</td>
<td>MT2-44:48, MT2-26:22, MT2-20:05, IT2a-11:13, MT3-23:57, PM5-04-2:33, CM1_2-21:25, CM1_2-20:10a, MT1-34:42, DE1-08, DE1-01, PM4-07-4:24, PM2-61:01</td>
</tr>
<tr>
<td>contextuality.</td>
<td>The situated aspects of an activity domain</td>
<td>PM4-06-2:29a, PM4-06-2:29b, PM4-06:4-04, PM4-08-3:55, PM2-45:43, PM4-07:1-12, VDR1-059:22, VDR1-061:43, VDR1-062:16a, VDR1-063:08, VDR1-071:03, PM2-00:18, PM1-36:01, MT4-19-2:05b</td>
</tr>
<tr>
<td>coordination and control.</td>
<td>Issues related to coordination and control as in the white space in the PROPS project model</td>
<td>PM4-05-1:07, PM4-07-4:24, PM4-06-2:29a, PM1-17:35, PM3-70:46c, PM3-70:46a, PM3-70:46b, PM3-08:07, PM3-23:15, IT2a-21:43b, ERI-1994-04-01, VDR1-078:45a, PM3-41:40b, MT4-14-2:47, MT4-13-4:48b, MT4-10-1:27, MT4-09-3:23, MT4-09-0:1a</td>
</tr>
<tr>
<td>credit.</td>
<td>Getting credit for a good job done</td>
<td>PM2-52:29</td>
</tr>
</tbody>
</table>

256
Table 36. The grounding of effect categories

<table>
<thead>
<tr>
<th>Category</th>
<th>Category description</th>
<th>Grounded in</th>
</tr>
</thead>
<tbody>
<tr>
<td>curiosa,</td>
<td>issued related to all kinds of remarkable things</td>
<td>IT2b-24:56, IT2a-06:40</td>
</tr>
<tr>
<td>customer,</td>
<td>all kinds of customers</td>
<td></td>
</tr>
<tr>
<td>daily tuning,</td>
<td>Tuning of the eMatrix implementation</td>
<td>VDR1-055:10, VDR1-015:42b, MT4-13-4:48b</td>
</tr>
<tr>
<td>data consistency,</td>
<td>consistency of data, e.g. no copies of the same data item</td>
<td>MT4-09-2:41, MT3-54:16, MT3-38:58, MT3-19:11, MT3-40:42, PM3-49:57, PM3-65:36</td>
</tr>
<tr>
<td>decentralization,</td>
<td>issues related to distribution of e.g. projects</td>
<td>PM2-02:32, PM2-33:13a</td>
</tr>
<tr>
<td>dependencies,</td>
<td>All kind of dependencies, closely related to traceability</td>
<td>PM3-41:40a, VDR1-082:21, PM4-07-0:37, MT3-52:20, MT3-50:55, CM1_1-23:55b, CM1_2-20:10a, MT1-24-14, MT3-70:13, MT4-09-0:01a, MT3-60:01, MT3-19:11, CM1_2-24:49, CM1_2-10:05, PM4-05-3:33</td>
</tr>
<tr>
<td>deployment,</td>
<td>issued related to deployment of the FWK e.g. in a project</td>
<td>PM4-14-0:28, PM4-07-2:17, ERI-2002-08-01:2, ERI-2002-08-01:3, VDR1-019:35, VDR1-023:12, VDR1-035:05, VDR1-037:26, VDR1-038:28a, VDR1-038:28b, VDR1-042:53, MT2-11-10a, PM2-22:25, PM2-41:45, PM2-54:07, PM3-10:10, PM3-25:03b, PM3-28:14, PM3-29:34, PM3-48:57, MT2-11:10a, MT2-52:54, MT2-47:48, MT2-24:10, MT3-76:00, MT3-33:02, MT3-29:41, MT3-29:05, MT3-09:55, MT4-02-3:24, MT4-03-0:20, CM1_1-23:55a, CM1_2-14:13, MT1-16:45, MT1-15:42, ERI-2000-06-08, ERI-2000-01-05, ERI-1999-03-13, ERI-1998-07-01, VDR1-053:18, VDR1-052:41, PM3-13:38, CM1_1-26:11b</td>
</tr>
<tr>
<td>deviations,</td>
<td>categories related to deviations from the ADT concept, for example in MARS</td>
<td>PM4-05-2:13, PM4-07-1:12, PM4-07-2:17</td>
</tr>
<tr>
<td>early access,</td>
<td>Early access to information in cooperating activity domains</td>
<td></td>
</tr>
<tr>
<td>effort,</td>
<td>issues related efforts, manhours</td>
<td>VDR1-078:45c, VDR1-078:45b, VDR1-078:45a, PM4-08-1:15, PM4-06-4:04, PM4-05-4:52b, PM3-14:28, MT4-10-1:27, MT3-19:11, CM1_1-23:40a, CM1_2-23:55a, CM1_2-14:50b</td>
</tr>
<tr>
<td>entrepreneur,</td>
<td>issues related to entrepreneurs driving an issue they believe in, often against established opinions</td>
<td>PM2-00:18, PM2-18:22, PM2-57:10, PM2-57:46b, PM3-84:39, PM3-11:00, PM3-28:14, PM3-31:31a, PM3-31:31b, PM3-34:41a, IT2a-75:07</td>
</tr>
<tr>
<td>establishment,</td>
<td>The DCS is divided into an establishment and expansion phase. During the establishment and exploration and consolidation is taken place. Not until the consolidation is stable, an expansion may take place. During consolidation, a daily tuning of the application may take place.</td>
<td>VDR1-053:18, VDR1-044:12a, VDR1-031:35, PM3-80:02b, PM3-80:02a, CM1_1-28:51</td>
</tr>
<tr>
<td>evolution,</td>
<td>Evolution: gradual changes in the AD</td>
<td>VDR1-028:26b, VDR1-024:26a, PM4-01-4:35, PM4-04-45:43, ERI-2002-08-01:3, PM4-09-0:55, PM3-75:26c, PM3-17:09, PM3-54:08a, PM3-57:03b, PM3-60:46, IT2b-18:07, IT2b-15:47a, IT2a-102:22, IT2a-97:16a, IT2a-09:18b, MT3-62:32, MT3-35:54, MT3-29:05, MT3-14-3:48b, MT4-12-0:59a, MT4-07-2:05, CM1_2-29:40a, CM1_2-26-47, CM1_2-16:11b, CM1_2-07:57, CM1_2-06:56, CM1_2-02-68, MT4-29:07, MT4-19:61, MT3-31:38, MT1-29:50, MT1-12-57, DE1-02, CU1-4, CU1-1, ERI-2000-02-14, ERI-2000-01-31, VDR1-052:41, PM3-48:57, PM3-17-45, PM3-11:56</td>
</tr>
</tbody>
</table>
Table 36. The grounding of effect categories

<table>
<thead>
<tr>
<th>Category</th>
<th>Category description</th>
<th>Grounded in</th>
</tr>
</thead>
<tbody>
<tr>
<td>exploration</td>
<td>Issues related to the exploration phase in DCS</td>
<td>VDR1-020:41a, VDR1-014:18, VDR1-012:27, VDR1-009:26a, PM4-14-0:28, PM4-13:3:12b, PM2-32:14, PM2-20:10, PM4-07-2:17, PM3-60:02b, PM3-47:02b, MT3-35:00, MT3-29:41</td>
</tr>
<tr>
<td>focal change</td>
<td>Issues related to focal changes</td>
<td>PM4-06-4:04, PM4-06-2:29b, MT4-14:2-47</td>
</tr>
<tr>
<td>focus</td>
<td>issues related to focusing efforts on one target</td>
<td>PM4-06-4:04, PM1-15:04b, PM1-17:35b, MT4-05-4:33b, PM5-04-2:33, MT1-21:09, MT1-19:25, DE1-01</td>
</tr>
<tr>
<td>fun</td>
<td>issued related to having fun at work</td>
<td>CM1-2:29:40b, CM1-2:05:40, CM1-2:03:29, CU1-5</td>
</tr>
<tr>
<td>generalization</td>
<td>indications that the FWK or the ADT may be generalized</td>
<td>PM4-12-3:43a, ERI-2002-07-01:1, PM3-23:15, PM3-41:40b</td>
</tr>
<tr>
<td>global</td>
<td>Global aspects</td>
<td>MT2-50:54, MT2-47:46, MT2-40:16a, MT2-29:34c, MT2-26:22, MT2-20:05, MT3-38:58, MT3-19:11</td>
</tr>
<tr>
<td>good enough</td>
<td>issued related to good enough, lagom, a canonica way of working</td>
<td>MT4-18-4:46, MT4-13-4:48a, MT4-13-1-56b, MT4-13-1-56a, MT4-12-4:18, ERI-1994-12-01</td>
</tr>
<tr>
<td>history</td>
<td>issued related to the history of the FWK</td>
<td>VDR1-002:16, VDR1-040:31, VDR1-044:12b, VDR1-050-47a, VDR1-050-47b, VDR1-026:58, PM2-10:50a, PM2-18:22, VDR1-049:21, PM2-20:10, PM3-82:01, PM3-05:56, PM3-07:54, PM3-09:07, PM3-11:00, PM3-13:08, PM3-17:45, PM3-25:03a, PM3-29:34, PM3-41:40a, VDR1-003:42, MT2-24:10, CM1-2-01:33, MT1-10:15, ERI-1999-08-25</td>
</tr>
<tr>
<td>human relations</td>
<td>human relations (HR), how people interact, work together etc. These are the soft issues in working together, something I really don't adress in the thesis.</td>
<td></td>
</tr>
<tr>
<td>iFame</td>
<td>issues related to the iFame project</td>
<td>ERI-2001-06-06b, ERI-2000-11-16</td>
</tr>
<tr>
<td>immutable mobile</td>
<td>Issued related to immutable mobiles. This is a concept in ANT which indicates a network with strong properties of irreversibility and effects which transcends time and space, e.g. a standard.</td>
<td>VDR1-063:06, PM2-57:46a, PM3-85:57, PM3-54:08b, IT2b-09:53, IT2a-102:22, IT2a-50:38, IT2a-34:37b</td>
</tr>
</tbody>
</table>
Table 36. The grounding of effect categories

<table>
<thead>
<tr>
<th>Category</th>
<th>Category description</th>
<th>Grounded in</th>
</tr>
</thead>
<tbody>
<tr>
<td>in-house development</td>
<td>issues regarding in-house development of tools etc. or sourcing them from outside</td>
<td>IT2a-25:10, ERI-2002-10-19, ERI-1994-06-14</td>
</tr>
<tr>
<td>incremental</td>
<td>Step-wise development</td>
<td>VDR1-007:34, PM3-41:40b, IT2a-21:43b, MT4-18:10b, ERI-1999-12-01, ERI-1997-10-23, ERI-1996-04-04</td>
</tr>
<tr>
<td>information model</td>
<td>The conceptual model referred to as information model, business model, object model, product model etc.</td>
<td>VDR1-074:37, VDR1-071:03, VDR1-066:35, VDR1-063:06, ERI-2002-08-01:2, ERI-2002-08-01:3, PM-12-0:03a, PM-11-1:20a, PM-11-1:20b, PM-05-5:2-13</td>
</tr>
<tr>
<td>infrastructure</td>
<td>issued related to infrastructure like the Intranet, routers, PC's etc</td>
<td>VDR1-035:05, VDR1-040:31, VDR1-044:12a</td>
</tr>
<tr>
<td>innovations</td>
<td>issued related to all kinds of innovations</td>
<td>VDR1-006:36, VDR1-007:34, RES1-01, VDR1-044:12b, VDR1-047:30, VDR1-055:10, VDR1-087:48</td>
</tr>
<tr>
<td>interaction</td>
<td>Issued related to all kinds of interactions</td>
<td>PM4-05-2:13, PM4-05-4:52a, PM4-06-1:13a, PM4-06-4:04, PM4-07-1:12, VDR1-099:27b, PM1-24:55</td>
</tr>
<tr>
<td>IS - ClearCase</td>
<td>issues related to ClearCase</td>
<td>PM3-82:01, PM3-75:26c, PM3-05:56, ERI-2000-01-12</td>
</tr>
<tr>
<td>IS - ClearDDTS</td>
<td>issues related to ClearDDTS</td>
<td>PM3-54:08b, PM3-54:08c</td>
</tr>
<tr>
<td>IS - CMTool-Project</td>
<td>issues related to CMtool Project</td>
<td>PM2-10:50a</td>
</tr>
<tr>
<td>IS - DOORS</td>
<td>issues related to DOORS</td>
<td>PM2-10:50b</td>
</tr>
<tr>
<td>IS - Information Systems</td>
<td>issues related to tools other than eMatrix</td>
<td>ERI-1994-04-26</td>
</tr>
<tr>
<td>IS - Metaphase</td>
<td>issues related to Metaphase</td>
<td>IT2b-28:58, IT2b-09:53, IT2a-97:16a, IT2a-77:25a, IT2a-60:59b, IT2a-60:59a, IT2a-54:08b, IT2a-54:08a, IT2a-00:38, IT2a-42:42, IT2a-36:02, IT2a-34:37b, ERI-2001-04-09, ERI-1997-11-10, ERI-1994-12-16</td>
</tr>
<tr>
<td>IS - ProActive</td>
<td>issues related to ProActive</td>
<td>PM2-04:24, PM2-02:32, PM2-01:12</td>
</tr>
<tr>
<td>IS - Rational</td>
<td>issues related to Rational tools</td>
<td>PM2-16:01a, ERI-2000-02-10, ERI-1998-07-07</td>
</tr>
<tr>
<td>IS - RequisitePro</td>
<td>issues related to RequisitePro</td>
<td></td>
</tr>
<tr>
<td>IS - TEPOS</td>
<td>issues related to TEPOS</td>
<td></td>
</tr>
<tr>
<td>IS architecture</td>
<td>The role of eMatrix in relation to other IS</td>
<td></td>
</tr>
<tr>
<td>IS federations</td>
<td>Issues regarding one database for eMatrix or several</td>
<td>PM4-11-1:20a, PM4-11-1:20b, PM-12-0:03b, PM4-12-0:03c, ERI-2002-08-01:1, ERI-2002-08-01:3, ERI-2002-08-01:4, ERI-2002-08-01:2, PM4-03-0:47, PM4-03-3:22, PM4-03-3:59, PM4-11-1:20c, PM4-12-0:03a, VDR1-074:37, VDR1-071:03, PM2-55:21, ERI-1995-01-27, ERI-1995-01-28</td>
</tr>
</tbody>
</table>
### Table 36. The grounding of effect categories

<table>
<thead>
<tr>
<th>Category</th>
<th>Category description</th>
<th>Grounded in</th>
</tr>
</thead>
<tbody>
<tr>
<td>Isomorphism,</td>
<td>issues regarding different world-views, phenomena- essence, common sense, everyday praxis</td>
<td>PM4-13:3:12, IT2a-06:40, DE1-08, CU1-2, MT4-18-4:49, MT4-04-0:54, MT3-82:13, MT3-79:29, CM1-2:33:09</td>
</tr>
<tr>
<td>Local development,</td>
<td>refers to local development of ADs (esp. tools) without coordination</td>
<td>IT2b-36:39, IT2b-30:23a, IT2b-13:17a, IT2b-09:53, IT2a-90:51, IT2a-77:25b, IT2a-60:59a, IT2a-04:08b, IT2a-43:38a, IT2a-42:42</td>
</tr>
<tr>
<td>Management support,</td>
<td>issues related to top management</td>
<td>VDR1-052:41, VDR1-094:24b, VDR1-094:24a, VDR1-092:56, PM2-16:01a, PM2-16:01b, PM2-18:22, PM2-43:00b, PM2-55:21, PM3-83:44, PM3-83:03, PM3-81:12, PM1-15:04b, PM1-17:35b, PM3-80:02a, MT2-40:16c, IT2a-60:59b, MT4-05:4-33b, MT4-05:1-05, MT4-05:0-10, ERI-2000-05-18, PM3-28:14</td>
</tr>
<tr>
<td>Market,</td>
<td>markets in contexts where the FWK was not present</td>
<td>PM1-40:23b, PM1-09:03b, PM1-09:03</td>
</tr>
<tr>
<td>Matrix functionality,</td>
<td>The functionality of Matrix</td>
<td>PM4-01:4:35, PM4-04-4:44, RES1-01, PM3-25:03b, 2000-04-20, ERI-2000-12-20, ERI-1998-02-04</td>
</tr>
<tr>
<td>Matrix training,</td>
<td>E. g. training how to use the Matrix</td>
<td>MT2-09:29, MT2-11:10b, PM4-04-1:48, PM4-04-3:39a, PM3-13:08, PM3-14:28, PM3-51:30a, MT2-11:10a, MT2-12:15, PM5-04-4:37b, PM5-04-4:37a, CM1-2:41:53</td>
</tr>
<tr>
<td>Matrix data entry,</td>
<td>The loading of data into Matrix</td>
<td>PM3-21:37, PM3-23:15, PM3-25:03a, PM3-25:03b, PM3-47:02a, PM3-48:57, PM3-57:33b, MT3-12:40, MT3-09:55, MT3-07:03, MT4-12:4-18, MT4-05:4-33b, MT4-05:4-33a, MT4-05:2-45, MT4-05:1-05, MT4-04-3:10</td>
</tr>
<tr>
<td>Matrix reports,</td>
<td>All kinds of reports from Matrix</td>
<td>PM3-59:53, MT4-19-2:05b, MT4-10-2:36, VDR1-078:45b</td>
</tr>
<tr>
<td>Matrix stability,</td>
<td>The stability of Matrix</td>
<td>CM1-2:29:40b, MT3-72:34</td>
</tr>
<tr>
<td>Matrix support,</td>
<td>Issues related to support</td>
<td>MT2-11:10a, MT2-60:54, MT2-47:46, MT2-40:16a, MT2-29:34c, MT2-20:05, MT2-17:38a</td>
</tr>
<tr>
<td>Need,</td>
<td>issues related to needs, e. g. for bringing in eMatrix as an IS</td>
<td>PM4-01-4:35, PM4-02-3:00, PM4-05-1:07, PM4-05-2:13, PM4-05-3:33, PM4-05-4:52b, PM4-06-1:13a, PM4-08-1:15, PM2-00:18, PM2-01:12, PM2-02:32, PM2-05:03, PM2-10:50a, PM3-07:54, PM3-09:07, PM3-10:10, PM3-21:37, PM3-23:15, PM3-36:42, PM3-41:43b, CM1_2-05:40, CM1_2-03:29</td>
</tr>
<tr>
<td>Negative tracks,</td>
<td>negative tracks in the AD, something is on its way to go wrong</td>
<td>PM2-33:13a, PM2-33:13b</td>
</tr>
</tbody>
</table>

260
Table 36. The grounding of effect categories

<table>
<thead>
<tr>
<th>Category</th>
<th>Category description</th>
<th>Grounded in</th>
</tr>
</thead>
<tbody>
<tr>
<td>new insights</td>
<td>issues related to new insights</td>
<td>PM4-08-3:55, VDR1-026:26a, RES1-01, VDR1-094:24b, PM2-36:01, PM2-41:45, PM2-47:39, PM2-61:01, PM3-64:39</td>
</tr>
<tr>
<td>one management tool</td>
<td>Issues related to one tool (Matrix) to support coordination</td>
<td>ERI-2001-08-24, MT3-19-11, MT3-40:42, MT3-54:16, MT3-50:55, MT3-65:01, MT4-14-2:47, MT4-09:0:01a, MT4-16-2:05, PM2-63:28, PM3-48:57, PM3-65:36, PM4-06-2:29a, PM4-07-4:24, PM4-06-2:29b, PM4-08-1:15, PM4-08-3:55, ERI-2000-05-17, ERI-2000-03-28, ERI-2000-02-10, PM3-54:16, MT3-38:58</td>
</tr>
<tr>
<td>organisational learning</td>
<td>Issues related to how the organisation learn including evolutionary issues</td>
<td>PM4-12-3:43b, RES1-01, VDR1-022:11, VDR1-052:41, VDR1-055:10, VDR1-083:30b, VDR1-078:45b, VDR1-083:30a, VDR1-085:23a, VDR1-085:23b, VDR1-087:48, VDR1-102:00, VDR1-102:00, PM2-06:36, PM2-20:10, PM2-23:56, PM2-47:39, PM2-60:25, PM3-65:57, PM3-81:12, PM1-37:12, PM1-35:08a, PM3-78:24a, PM3-67:26b, PM3-67:26a, PM3-14:26, PM3-21:37, PM3-23:15, PM3-28:14, PM3-36:42, PM3-47:02a, PM3-50:25, PM3-51:30a, PM3-60:46, ERI-2000-07-06</td>
</tr>
<tr>
<td>orientation</td>
<td>Issues related to orientation, see the whole picture, understand relations</td>
<td>MT3-50:55, MT4-16-2:05, MT4-14-2:47, MT4-09-4:57, MT4-09-3:23, MT4-09-0:01a, MT4-08-2:43, PM3-04-3:46, PM5-04-2:33, CM1_2-20:10a, MT1-27:43, MT1-24-14, DE1-14, DE1-06, DE1-04, ERI-2002-11-20</td>
</tr>
<tr>
<td>overview</td>
<td>overview of a complex situation</td>
<td>MT4-16-2:05</td>
</tr>
<tr>
<td>participation</td>
<td>issued related to actors participation in the activity. Also issues regarding credit for achievements</td>
<td>MT2-09:29, MT2-11:10b, VDR1-035:05, PM2-52:29, PM1-17:35, PM1-19:33b, PM1-27:31, PM1-27:31c, PM1-27:31b, PM1-44:08, PM1-35:06b, MT2-44:48, MT2-40:16a, MT2-21:42b, MT2-12:15, MT4-13-4:48a, MT4-13-1:56b, MT4-13-1:56a, MT4-12-4:18, MT4-06-2:51, MT4-65:4:33b, MT1-38:30, DE1-12, DE1-09</td>
</tr>
<tr>
<td>performance</td>
<td>tool performance in contexts where the FWK was not present</td>
<td>PM2-02:32, PM2-04:24</td>
</tr>
<tr>
<td>process - blueprint</td>
<td>Issues related to processes, blueprint: according to plan, opposite to evolutionary</td>
<td>ERI-1995-02-21, VDR1-020:41b, VDR1-011:15</td>
</tr>
<tr>
<td>process - RUP</td>
<td>The Rational Unified Process</td>
<td>PM3-41:43b</td>
</tr>
<tr>
<td>process - work package</td>
<td>issues related to the work package process - method</td>
<td>PM4-05-2:13, PM4-05-4:52a, PM4-06-4:04, PM4-07-1:12, PM4-12-3:43a, ERI-2002-07-01:1, PM4-05-3:33, PM4-06-1:13a, PM4-07-4:24</td>
</tr>
<tr>
<td>process</td>
<td>Issues related to processes, methods and concepts other than in the FWK</td>
<td>PM2-01:12, PM2-05:03, PM2-06:36, PM2-08:17, PM1-10:36, PM1-11:56, PM1-10:36b, PM1-15:04</td>
</tr>
<tr>
<td>product</td>
<td>concerns the object of the development task</td>
<td>PM1-32:10b, PM1-25:46, PM1-40:23b</td>
</tr>
<tr>
<td>project</td>
<td>categories related to the project as a whole</td>
<td>PM4-05-4:52b, PM1-17:35b, PM1-27:31c, PM1-37:12, PM1-24:55, PM3-06:42</td>
</tr>
</tbody>
</table>
CHAPTER

Table 36. The grounding of effect categories

<table>
<thead>
<tr>
<th>Category</th>
<th>Category description</th>
<th>Grounded in</th>
</tr>
</thead>
<tbody>
<tr>
<td>quality</td>
<td>categories related to the quality</td>
<td>VDR1-078:45c, PM4-06-4:04, PM4-05-4:52b, PM3-48:37, PM3-14:28, MT3-40-42, MT3-19-11, CM1-2-24:49, CM1-2-23:55b, PM3-16:53</td>
</tr>
<tr>
<td>quotation</td>
<td>quotations used in the dissertation</td>
<td>MT4-10-2:36, MT4-09-0:01a, MT3-55-52, MT3-35-40, CM1-33-09, CM1-2-24:40, VDR1-083:30a, PM4-08-3:55, PM4-07-4:24, PM3-48:57, PM3-31-40a, PM3-13-08, PM3-10-10, PM2-45-43, MT4-13-4:48b, MT4-14-2:47, MT4-16-0:25, MT4-16-2:05</td>
</tr>
<tr>
<td>responsibility</td>
<td>the pin-pointing of responsibility to the proper place</td>
<td>PM3-75-26a, PM4-06-2:29a, PM4-08-2:29b, PM4-08-4:04, PM4-08-3:55, PM3-75-26b, PM3-70-46d, PM3-70-46a, PM3-70-46b, MT4-09-0:01b, PM5-04-2:33, CM1-2-21-25, CM1-2-04-18</td>
</tr>
<tr>
<td>roles</td>
<td>All kinds of roles around the FWK</td>
<td>PM3-20-36, MT4-10-3:18, CM1-2-16:11a, DE1-06</td>
</tr>
<tr>
<td>scalability</td>
<td>issued related to scaleability</td>
<td>ERI-2002-08-01:4</td>
</tr>
<tr>
<td>signification</td>
<td>Signs are mediators between the human cognitive system and the environment of the individual.</td>
<td>PM4-07-0:37, VDR1-061:43, VDR1-078:45b, VDR1-085:23b, PM2-23:56, PM2-25-26, PM1-09-03, PM1-11:58, PM1-19-33, PM3-70-46c, IT2a-11-13, IT2a-09:18b, MT3-79-29, PM4-05-2:13</td>
</tr>
<tr>
<td>sine qua non</td>
<td>Necessary prerequisites</td>
<td>PM4-05-1:07, PM3-78-24b, PM3-10-10, PM3-41-40a</td>
</tr>
<tr>
<td>stabilizing core control</td>
<td>The control of the stabilizing core</td>
<td>IT2a-97:16a, MT3-64-52b</td>
</tr>
</tbody>
</table>
### Table 36. The grounding of effect categories

<table>
<thead>
<tr>
<th>Category</th>
<th>Category description</th>
<th>Grounded in</th>
</tr>
</thead>
<tbody>
<tr>
<td>stress, stress in contexts where the FWK was not present</td>
<td></td>
<td>PM1-22:55b, PM1-22:55</td>
</tr>
<tr>
<td>structure, structure in contexts where the FWK was not present</td>
<td></td>
<td>PM1-32:10c, PM1-32:10b</td>
</tr>
<tr>
<td>technical management, technical management in contexts where the FWK was not present</td>
<td></td>
<td>PM1-19:33</td>
</tr>
<tr>
<td>the unexpected, categories related to unforeseen events (sv. tillfälligheternas spel)</td>
<td></td>
<td>VDR1-042:53, VDR1-037:26, VDR1-035:05, VDR1-044:12b, VDR1-048:35, PM2-00:18, PM1-27:31b, PM1-21:33</td>
</tr>
<tr>
<td>traceability, Issued related to traceability</td>
<td></td>
<td>PM4-05:1-07, PM2-45:43, PM4-10:3-32, PM4-11-0:02, PM2-63:26, PM3-67:26a, PM3-57:03a, PM3-57:03b, MT3-60:01, MT3-58:23, MT3-56:52, MT4-06:2-43, PM5-04-3-46, PM5-04:2-33, ERI-2000-02-10</td>
</tr>
<tr>
<td>transcendence, The dialectics between tradition and renewal (Ehn)</td>
<td></td>
<td>PM3-64:35, MT4-11-1:49, MT4-11-0:27, CM1_2-08:21, ERI-2000-01-31</td>
</tr>
<tr>
<td>transparency, Issued related to transparency, to see how things are related.</td>
<td></td>
<td>PM4-05-2:13, PM4-06-2:29b, PM4-06-2:29a, PM4-07-0:37, PM4-07-4:24, PM3-75:26a, PM3-70:46d, PM3-70:13, PM3-67:26a, CM1_2-20:10a, MT3-60:01, MT3-56:52, MT3-50:55, MT3-19:11</td>
</tr>
<tr>
<td>tribute, Issues regarding getting tribute for things achieved</td>
<td></td>
<td>RES1-02</td>
</tr>
<tr>
<td>trust, categories related to trust</td>
<td></td>
<td>PM3-83:44, PM3-82:01, PM3-81:12, PM3-80:02a, PM3-78:24a, PM3-48:57, MT3-76:56, MT3-76:00, MT3-09:55, CM1_2-33:53, DE1-07</td>
</tr>
<tr>
<td>unsolved issues, Issues related to unsolved problems, a matter for further research.</td>
<td></td>
<td>PM4-08-3:55, PM4-09:0-55</td>
</tr>
<tr>
<td>vendor, Issued related to vendors</td>
<td></td>
<td>PM2-22:25, VDR1-032:37b, VDR1-035:05, VDR1-038:28a, VDR1-049:21, PM2-41:45</td>
</tr>
<tr>
<td>views, Issued related to different views of the information</td>
<td></td>
<td>CM1_2-16:11a</td>
</tr>
</tbody>
</table>
Index

A
action research 30
   intellectual framework 31
activity domain 51, 85, 93, 106
   definition of 84
Activity Theory 13, 18, 52–53, 84–85, 89
Actor Network Theory 52, 85, 87, 91, 119, 149
actor roles 158
anatomy 166, 218
artefact 34
authenticity 31

B
baseline 137, 157
bias 31

C
case study 28
Change Requests 137, 157
ClearCase 144, 147, 189
ClearDDTS 189
cognitive semantics 71
complexity 1, 8
   definition of 8
correlation 66
conceptual spaces 65
Construction PLanning tool 130–131, 133–
   135, 137, 144
coordination 6, 95–96
   definition of 98
   situations 99, 155
coordination constituents 12, 46, 72, 84, 120,
   127, 241
contextuality 12, 72, 74, 84, 122–123,
   125–127
domain transition 13, 72, 84, 103, 122–
   123
experiential learning 13, 72, 78, 127, 131,
   133, 135, 147
intersubjectivity 12, 72–73, 84
motive 13, 73, 85, 99
object 13, 73, 85
orientation 13, 72, 78, 84, 102, 127–128
outcome 13, 73, 85, 99
stabilizing core 13, 73, 80, 85, 103, 121,
   123, 125, 127, 146
temporality 13, 72, 79, 85, 102, 123–124
   tool usage 13, 73, 83, 85, 103
coordination domain 97
instanitiation 155
   A-domain 36, 144, 147–148, 152
   C-domain 36, 145, 147–148, 153,
   215
   L-domain 36, 143, 148, 152
   S-domain 35, 137, 144, 148
coordination items 6, 100
Corporate IS/IT 126, 133, 138, 142–143, 145,
   183–184, 189–190, 227
criticality 31
CRUD matrix 109

dialectical relation 56–57, 67
dialectics 46, 63, 115
DOORS 187, 194

eMatrix 112, 119, 130, 138–139, 148
Lean Client 141, 164, 178–179
empiri
   consequence category
   definition of 159
   effect category
   definition of 159
   grounding 159
Engineering Change Management 157

F
flexibility 136
Framework
   Conceptual Model 11, 106, 120, 129, 131,
   139, 145, 152–154
   Domain Construction Strategy 12, 106,
   114, 120, 151–153
   Information System 12, 106, 119–120,
   130–133, 137, 145, 148, 151–154
   Process Model 11, 106, 120, 129, 131–
   133, 148, 150–154
   Stabilizing Core 12, 106, 120, 123, 125,
   151
   Transition Model 12, 106, 120, 122, 150,
   152

G
Gaussian Adaptation 82, 238
generalization 33, 228
Grounded Theory 45, 159
axial coding 201
open coding 200
selective coding 204
grounding 33
U
UML 217
Unit of Analysis 29
usage domain 97
user resistance 164

V
validity 32
VHDL 122
  translators 122

W
waterfall model 19, 128
Supported by the Foundation for Advancement of Knowledge and Competence (KK-stiftelsen), the Industry Research School in Applied IT and Software Engineering at Linköping University offers a high-quality Ph.D. training programme exploiting Linköping’s leading position in the IT area. The research school has the capacity to admit more than twenty doctoral students affiliated with sponsoring companies. Thesis projects involve basic and applied research in the areas such as industrial software engineering, programming tools and systems, human-computer interaction and IT management.

This doctoral dissertation “A Framework for the Coordination of Complex Systems' Development” by Lars Taxén has been supported by the Industry Research School in Applied IT and Software Engineering at Linköping University, the Swedish National Board for Industrial and Technical Development, NUTEK, project P10518-1 and Ericsson AB.