Coordination Dynamics in Open-Source Based Platforms: “The Symbian Foundation Case”

Authors: Jesus Mascareño
          Imran Mohyuddin
ABSTRACT

Industry platforms, particularly open-source based platforms are emerging as the tipping point of a new trend of interorganizational relationships among firms. They are characterized by a large number of actors with different objectives that come and go. However in order to reap the benefits of network effects, reduce fragmentation and get access to a large pool of resources, coordination dynamics within the different actors to create and innovate the platform are needed.

As opposed to traditional literature where a single firm leads the evolution of the platform, a more democratic approach based on the institutionalization of coordination, the implementation of coordination processes and mechanisms is proposed. A study in the form of interviews and interactive forums was carried in the Symbian Platform, specifically in the Symbian Foundation to identify the main coordination dynamics.

The results showed that in the case of the Symbian Platform, firms’ first step towards coordination was to establish the formal structure of coordination, in this case the Symbian Foundation. Consequently the Symbian Foundation established the processes and coordina-
tion mechanisms by which all of the actors participate and access to a pool of resources. The study describes the evolution from democratic coordination to an increasing self-coordination promoted by the Symbian Foundation within its members.

**Key-Words**
Open-Source Based Platforms, Coordination, Interorganizational Relationships, Coordination Mechanisms, Coordination Processes
Acknowledgements

The acknowledgment provided us an opportunity to express our gratitude to people who stood besides us during this research and whole SMIO program.

First we would like to thank Per Aman for providing us with guidance and fruitful thoughts during the entire thesis. We would also like to thank Ramsin Yakob for his special guidance during this thesis, those short meetings made the difference in linking the ideas about the topic and research area.

This thesis wouldn’t be possible without the support from the Symbian Foundation’s staff, particularly from Lauren Sarno, Head of Members from the Symbian Foundation, therefore in this space we want to express our gratitude to them.

Jesus Mascareño Acknowledgements:
I would like to thank my wife Lizbeth, for showing me the road in the first place and always showing me the bright side of the things, this is our accomplishment. I also would like to thank to my parents Rodolfo and Teresita, my mother and father in law Dolores and Benjamin for their unconditional support.

Imran Mohyuddin’s Acknowledgements:
I would like to thank my parents (G. M. Tahir and Shama Mohyuddin) for their unconditional support, both financially and emotionally throughout my program. In particular, the patience and understanding shown by my mum, dad, uncle, brother and sisters during my study years is greatly appreciated.
Contents

1 Introduction ...................................................................................................................... 1
  1.1 Problem Area ............................................................................................................. 3
  1.2 The challenge ............................................................................................................. 7

2 Theoretical Framework and Core Ideas ............................................................................... 9
  2.1 Industry Platforms ..................................................................................................... 9
    2.1.1 Platform Actors .................................................................................................. 10
    2.1.2 Two-sided and Multi-sided Platforms ................................................................. 11
    2.1.3 Open-Source Based Platforms ............................................................................. 12
    2.1.4 Platform Leadership .......................................................................................... 15
  2.2 Coordination Theory ................................................................................................... 18
    2.2.1 Institutionalization of Coordination ..................................................................... 19
    2.2.2 Coordination Processes ...................................................................................... 23
    2.2.3 Coordination Mechanisms .................................................................................. 30

3 Methodology .................................................................................................................. 35
  3.1 Research settings ....................................................................................................... 35
  3.2 Research methodology ............................................................................................... 35
  3.3 Research Approach ................................................................................................. 36
  3.4 Research Strategy ..................................................................................................... 37
  3.5 Data collection methods ........................................................................................... 37

4 Empirical Study .............................................................................................................. 40
  4.1 History of Symbian ................................................................................................... 40
    4.1.1 Early Alliances ................................................................................................. 40
    4.1.2 Platform Formation .......................................................................................... 41
  4.2 Organization ............................................................................................................... 45
    4.2.1 The Core ............................................................................................................ 47
    4.2.2 Symbian Groups ............................................................................................... 51
    4.2.3 Operations ........................................................................................................ 52
    4.2.4 Licensing Symbian ........................................................................................... 54
  4.3 The Platform Release Process .................................................................................... 54
    4.3.1 Symbian Release Planning ............................................................................... 57
  4.4 Interaction Channels ................................................................................................ 60
4.4.1 Online Ideas Exchange ................................................................. 60
4.4.2 Symbian Horizon ................................................................. 62
4.4.3 The Symbian Blog ............................................................... 64
4.4.4 Symbian Developer ............................................................ 64
4.4.5 Symbien Signed ............................................................... 65

5 Analysis...................................................................................... 66
  5.1 Structural Composition of the Symbian Platform ..................... 66
  5.2 Coordination Processes ....................................................... 69
  5.3 Coordination Mechanisms ................................................... 71
  5.4 Interpretive Model ............................................................... 73

6 Conclusion ................................................................................ 76

7 Limitations ................................................................................ 78

8 Implications ............................................................................. 79
  8.1 Implications for Practice ..................................................... 79
  8.2 Implications for Theory ....................................................... 79
  8.3 Implications for further research ......................................... 79

9 References: .............................................................................. 81
List of Figures:

**Figure 1:** Vertical Disintegration of the PC Industry (Adapted from Grove, 1996) ......................... 4

**Figure 2:** Industry Platform ............................................................................................................. 10

**Figure 3:** Institutionalized Coordination .......................................................................................... 22

**Figure 4:** Virtuality ............................................................................................................................ 24

**Figure 5:** The Research Process Onion (Saunders et al. 2003) .......................................................... 36

**Figure 6:** Composition of the Symbian Foundation ........................................................................... 46

**Figure 7:** The System Model (www.symbian.org) ............................................................................. 50

**Figure 8:** Leadership Team Chart (Adapted from: www.symbian.org) ............................................... 54

**Figure 9:** Platform Release Process (Adapted from: www.symbian.developer.org) ......................... 55

**Figure 10:** The Road Map (Adapted from: www.symbian.developer.org) .......................................... 58

**Figure 11:** Integration Plan Burndown Chart (Adapted from: www.symbian.developer.org) ............ 59

**Figure 12:** Sample of the Integration Plan’s “Two-week heartbeat” (Adapted from: www.symbian.developer.org) ....................................................................................................................... 59

**Figure 13:** The Innovation Life Cycle of Ideas at Symbian Foundation .......................................... 61

**Figure 14:** The Coordination Phases Model ....................................................................................... 74
1 Introduction

This study blends technology and interorganizational ties from a strategic perspective and it focuses in understanding the coordination dynamics within agglomerations of firms, identifying the mechanisms to achieve inter-organizational goals and the strategies placed to deal with a scenario of complex inter-organizational ties. It is intended to analyze these agglomerations from an industry platform perspective, particularly from an open-source based platform perspective. Focusing on open-source based platforms enables an extended analysis of increasing complex and difficult to coordinate agglomeration of firms.

The emergence of platforms as building blocks that act as engines of innovation and redefine industrial architectures, is a novel phenomenon affecting most industries today, from products to services. (Gawer, 2009). Moreover Evans et al (2006) argues platforms are at the center of the tectonic industrial change now business arena is facing. Industry platforms are found at the industry level and its configuration has been widely used especially in high technology industries. Gawer (2009) described industry platforms as building blocks (they can be products, technologies or services) that act as a foundation upon which an array of firms can develop complementary products, technologies or services. Gawer and Cusumano (2002) mentioned that industry platforms provides a common foundation or core technology that a firm can reuse in different product variations, an industry platform provides this function as part of a technology “system” whose components are likely to come from different companies which are called “complementors.”

In software development, opening the platform requires the use of an open source, which is available to anyone for modifications and the distribution of this improved version (Goldman and Gabriel, 2005). In open source based platforms the core element is open without restrictions to any firm willing to participate in the platform; it can alter it to its benefit (Dedrik and West, 2003).
However the openness of open-source based platforms stimulates its complexity in terms of number of actors, which in turn increases the network effects. These effects lead a single change in a module to affect the entire system (Gawer and Cusumano, 2002). Therefore open-source based platforms challenges the synchronization of innovation in the system as a whole, as Adner and Kapoor (2006) argued focal firms within platforms may be dependent on external actors in its ecosystem overcoming their own innovation challenges and the existence of ecosystem bottlenecks limiting the innovation process within networked firms.

This complexity limits the communication and coordination in platforms as Goldman and Gabriel (2005) argued the most critical problems for open-source strategies is the duplication of efforts in developing a module of the platform, the control efforts by groups of interest within the platform and most importantly the belief that the community will arise by itself. Therefore stand alone strategies don’t work when a company depends on the collective health of the organizations that influence the creation of a product (Lansiti and Levien, 2004).

The deficient coordination of innovation among the different actors affects how the entire platform develops, for example the applications for mobile phone software platforms has taken longer to take off than for personal computers, video game consoles, or PDAs as Evans et al (2006) explained “the mobile phone software platforms lack the ability to control the ecosystem and drive innovation in it”.

Complexity of open-source based platforms and the fragmentation of once heterogeneous markets (Datamonitor Study on Technology, 2004) makes difficult to achieve shard goals and coordination among its members. Therefore a fundamental challenge of managers of firms in industry platforms now face is the management of assets that are no directly owned by their own firms (Lansiti and Levien, 2004) and understanding coordination mechanisms that go beyond following a platform leader but to actively participate in the design of the innovation path of the platform.

Traditional models of strategy that emphasize internal competencies fail to account for coordination dynamics within this new agglomeration of firms because they focus on the evo-
lution of firm capabilities and business models instead of on the relationship between the firm and its external ecosystem (Levien and Lansiti, 2004). The literature discussing platforms has concentrated in platform leadership as coordination strategy however given that open-source based platforms are developed in a rapid change environment where uncertainty must be tackled by its members in a coordinated way the platform leadership view does not apply for this complex scenario.

### 1.1 Problem Area

In the following section a review of the environmental pressures and shifts in interorganizational ties will be discussed to provide a picture of the landscape in which firms in open-source based platforms interact and the challenges this scenario bring to its coordination efforts.

Recently an industry phenomenon has been taking form and is affecting how we conceive the structure of markets. We are now seeing agglomerations of small- and medium-sized firms in industries characterized by volatile or rapidly shifting demand, each firm specializing in a particular phase of production or in a particular production process. Finished goods are produced by groups of firms collaborating in rapidly shifting constellations (Gilson et al, 2009).

Traditional models of strategy that emphasize internal competencies fail to account for dynamics within this new agglomeration of firms because they focus on the evolution of firm capabilities and business models instead of on the relationship between the firm and its external ecosystem (Levien and Lansiti, 2004). An affluent group of researchers have taken the task to explain this phenomenon, the outcome is a diverse group of definitions which sometimes overlap or interchange without notice. In the following section a review of these concepts will be done in order to identify commonalities across definitions and the subject of study will be appointed.

Industries increasingly have become more complex as they evolve, what generally happens is that a once firm’s vertically related businesses break down into independent firms (Chen,
2005) increasing the number of actors involved in the production of a product or service. These specialized firms emerge to develop certain components of a larger puzzle once produced by a small group of firms (Gawer and Cusumano, 2002). Stigler (1951) refer to this effect as a common characteristic of evolving industries “vertical disintegration is the typical development in growing industries, vertical integration in declining industries”.

In a depth analysis on the evolution of the PC Groove (1996) and then Baldwin and Woodward (2009) mapped how the computer industry moved from a vertical integrated industry to a modular cluster or horizontal integration (see figure 1). In figure 1 we can see how the PC evolved from the traditional “vertical silos” to “modular clusters” integration and how this shift in the integration of the PC industry meant an increase in the complexity of the industry.

**Figure 1:** Vertical disintegration of the PC Industry (adapted from Grove, 1996)

As the environment’s complexity increases the actor’s dependency within the each other increases as well. This dependency is augmented due to end users are interested in working systems, not unconnected components, but whether they obtain those systems from one firm or from several (Evans, Hagiu and Schmalensee, 2006). Therefore firms aiming to improve the overall product or service must relay in external actors, as Adner and Kapoor (2006) argued focal firms within ecosystems may be dependent on external actors in its ecosystem to overcome their own innovation challenges. This dependency tightens firms into
relationships with firms who serve common end users. Firms are not islands but are linked together in patterns of co-operation and affiliation (Richardson, 1972).

The result of these developments is the agglomeration of interconnected firms working together intentionally or unintentionally to serve a common end user. In an attempt to understand the phenomenon authors first situated this event as the expansion of the barriers of a focal company. These attempts lead researchers to term the phenomenon “metacorporation” or “Moebius-strip organization” (Sabel 1991), “boundaryless organization” (Ashkenas et al. 1995), “virtual corporation” (Davidow and Malone 1992), “dynamic network” (Miles and Snow 1986), and network organization (Baker, 1992). However Lomi (1997) argued this new form is more open toward its environment because it is so deeply embedded in networks of exchange with other organizations and therefore it is difficult to distinguish its inside from its outside.

In explaining this phenomenon Rosenkoft and Tushman (1998) situated firms in community networks or inter-organizational networks that are characterized by complex technologies. Lansiti and Levien (2004) described these agglomerations overlapping the terms business networks and business ecosystems. In their definition business ecosystems are characterized by a large number of loosely interconnected participants who depend on each other for their mutual effectiveness and survival. In this scenario, the challenge this firms face is the management of assets that are not directly owned by their own firms.

There has been a trend explaining this event under the term of networks (Glambos and Sewell, 1995; Prencipe et al, 2003; Jarillo, 1988; Imai and Baba, 1989; Ahuja, 2000). Jarillo (1988) defined these agglomerations as strategic networks, which are a long term, purposeful agreements among distinct but related for-profit organizations that allow firms in them to gain or sustain competitive advantage vis-à-vis their competitors outside the network. Imai and Baba (1989) described network as an organization having a core firm both strong and weak ties with constituent members, that is, other firms, research centres and universities. In explaining the benefits for participating in a network Ahuja (2000) argued network present two major benefits: first, resource sharing that enables firms to combine knowledge, skills, and physical assets. Second, access to information spillovers, network relation-
ships act as information conduits through which news about discoveries and failed approaches are exchanged.

In a study on software development Carliss, Baldwin and Clark (1997) argued the existence of modular clusters composed of hundreds of companies in a constantly innovating industry. According to the authors a modular system is composed of units (or modules) that are designed independently but still function as an integrated whole. Modularity requires companies to understand products at a deep level and be able to predict how modules will evolve.

The phenomenon also has been termed as systems (Prencipe et al, 2003; McCormack, 2001; Ethiraj and Levinthal, 2002; Schilling 2000), Hobday et al (2003) identified complex systems of firms around the production of airplanes, automobiles, and electric power system. They are complex in the sense that each system has a number of different components or elements, and for effective performance all have to work together. The authors highlight the importance of external activities of firms in complex systems, as they integrate components, skills, and knowledge from other firms, including suppliers, users, and partners, in order to deliver ever more complex products and systems. Powell and Grodal (2005) highlighted complex networks of firms, universities, and government labs as critical features of many industries, especially in fields with rapid technological progress.

In another attempt to explain this trend has been discusses under the label of industry platform, Gawer (2009) described industry platforms as building blocks (they can be products, technologies or services) that act as a foundation upon which an array of firms can develop complementary products, technologies or services. In the same line of thought Gawer and Herderson (2007) argued that high-technology industries offer products or services which can be described as systems of interdependent components, built around or on top of platforms.

In a more broad definition Adner (2006) place firms in interconnected ecosystems, this ecosystems allow firms to create value that no single firm could have created alone. Strategy making in an ecosystem is iterative—it has to be, because there are so many interconnected
Coordination Dynamics in Open-Source Based Platforms

pieces and players. The challenge in an interconnected ecosystem is to develop a tentative agreement on the performance expectations that would constitute success.

As it was observed above, the agglomerations of firms is a result of the vertical disintegration of industries. In this scenario product or services are provided by a group of specialized firms who work together to serve a common end user. Specialized firms depend on each other for their mutual effectiveness and survival given that end users look for working systems not individual components of a whole. Therefore in order to thrive in an interconnected agglomeration firms must interact with the different actors to integrate skills, information and knowledge from other firms. Given this characteristics of dependency, the challenge is to define success and coordinate efforts to reach it.

All of the definitions discussed previously attempted to describe the same phenomenon, they are a valid effort to understand the unfocused, unbounded, and constantly evolving networks (Lansiti and Levien, 2004). However for the purpose of this study, platforms, more specifically open source based platforms will be used as the focal point of study.

1.2 The challenge

This study is derived from the necessity to identify coordination dynamics in open-source based platforms. The relevance of this necessity lies in the fact that open-source based platforms present a complex composition, no barriers to external actors, increasing market fragmentation along the platform and where in the lack of coordination firms get hold of standalone strategies. Given these developments the purpose of this study is to:

Understand how unfocused and constantly evolving agglomeration of firms with different objectives, technologies, information and resources coordinate to create an open-source based platform.

In order to reach the study’s purpose, shedding light over the following questions is needed:

1. Who promotes, assures and regulates coordination in an open-source based platform?
2. What are the coordination dynamics that focus contribution and ties together an open-source based platform?
   a. Under which interorganizational process firms will coordinate to create an open-source based platform?
   b. Which mechanisms are placed to channel efforts towards a common goal?

The answer to these questions have implications for theory and practice, in practice it will help managers of firms who are active in platforms or are exploring the possibility to adhere to a platform to generate awareness of the imperative of coordination among these interorganizational form, identify tools that enhance and generate coordination processes and lastly to reap benefits from networks effect in complex industry platform. In theory this study will close the gap of how interactions within members of an open-source based platform are coordinated.
2 Theoretical Framework and Core Ideas

In this chapter theory on industry platforms and open-source based platforms will be covered to provide with a descriptive view of the platform phenomenon, its components, and characteristics. Then a review on early attempts to explain coordination under the concept of platform leadership is carried. A review on interorganizational coordination theory is undertaken in order to propose a coordination model based on structures, processes and mechanisms.

2.1 Industry Platforms

Industry platforms are found at the industry level and its configuration has been widely used especially in high technology industries, Gawer (2009) described industry platforms as building blocks (they can be products, technologies or services) that act as a foundation upon which an array of firms can develop complementary products, technologies or services. Lansiti and Levien (2004) describe industry platforms as services, tools, or techniques that other members of the ecosystem can use to enhance their performance. While Gawer and Henderson (2007) argued high-technology industries offer products or services which can be described as systems of interdependent components, built around or on top of platforms.

Industry platforms differ from product platform in the sense that product platform occurs at in-house level while industry platforms occur at an inter-firm level. In order to clarify this Gawer and Cusumano (2008) sustained that a product is largely proprietary and under one firm’s control, whereas an industry platform is a foundation technology or service that is essential for a broader, interdependent ecosystem of businesses.

Similar to product platform, Gawer and Cusumano (2002) mentioned that industry platforms provide a common foundation or core technology that a firm can reuse in different product variations, an industry platform provides this function as part of a technology “system” whose components are likely to come from different companies which are called “complementors.” The industry platform presents a strong dependency between the core technology and its complementors, in this sense Gawer and Cusumano (2002) pointed that
industry platforms have relatively little value to users without these complementary products or services. And they suggest that not all products, services or technologies can become platforms: to have platform potential, they must (1) perform a function that it is essential to a technological system and (2) it should be easy to connect to or build upon to expand the system of use as to allow new and even unintended end-uses.

Based on the discussion above it can be concluded that industry platform in its simplest form contains two main elements in its composition, one the existence of a stable or low variability component, which is a firm who owns a product, technology or service on which high variability or complement components are combined to deliver an even complex product, service or technology. Based in this a graphical interpretation has been drawn (see figure 2).

![Figure 2: Industry Platform](image)

### 2.1.1 Platform Actors

In industry platforms two main actors can be identified, core components or platform owners and platform’s complementors. Gawer and Henderson (2007) defined platform “owner” as a firm that owns a core element of the technological system that defines its forward evolution. The authors mentioned that platform complementors are the suppliers of the platform owner, which together they create a complementary market, for example McAfee is Microsoft’s complementor. In industry platforms’ complementors interact directly with the demand-users using the same platform as baseline, therefore complementors provide end-user with the complementary components of the platform.
At the beginning of the industry platform life, the main task is to attract complementors in order to create a strong network and create a niche for the core platform. Whitney et al. (2004) related the starting phase of a platform with an interactive search by users and designers for requirements and matching architecture. Gawer and Cusumano (2008) mentioned that platform owners can emerge as platform leader and work with the companies supplying complementary products and services to form an ecosystem of innovation that can greatly increase the value of their innovations as more users adopt the platform and its components.

An important benefit of the platform is the innovative environment that evokes in its complementors by promoting competition among complementors and different configurations. In this sense Gawer (2009) noted that in industry platforms, the use of end products or services is not predetermined, which leaves open the door for complementary innovation to determinate the possibilities of the platform.

### 2.1.2 Two-sided and Multi-sided Platforms

According to the number of interdependent customers industry platforms can be two-sided or multi-sided. However this concept has evolved to include more elements for example Roson (2005) described two-sided markets as two (or more) parties interacting on a platform, and the interaction is affected by special “indirect” network externalities. Rochet and Tirole (2001) mentioned that two-sided platforms are the ones in which economic value is created by interactions or transactions between pairs of end users, buyers and sellers and such transactions are mediated by a platform. Chakravoty and Roson (2004) provided a deeper description of two-sided markets by defining them as a platform(s) providing goods and services to two distinct end-users where the platform attempts to set the price for each type of end-user to “get both sides on board” because the benefits of one type of end-user increases as the participation of the other type of end-user increases.

Two-sided markets are characterized by a special type of network externality. This externality does not depend on consumption of agents in the same class (for example, consumers of the same product), but on consumption of different, but “compatible”, agents on an opposite market side. Network externalities exist whenever the matching quality improves when
more alternatives become available and more interactions are possible if more partners are available. (Roson, 2005).

Two-sided platforms can be found in the credit cards, Youtube, Facebook and Yellow Pages. In each of these examples the platform owner has to attract both sides of the market in order to provide value for all the actors. The video game platform is conformed of video game consoles (platform owner), video game developers (complementors) and consumers, for video consoles platform to be attractive it must promote the development of video games and users of the platform (console).

According to Evans (2003) multi-sided platforms coordinate the demand of distinct groups of customers who need each other. Hagiu (2009) mentioned that multi-sided platforms provide a support that facilitates interactions (or transactions) among the two or more constituents (sides) that it serves, such that members of one side are more likely to get on board the MSP when more members of another side do so. A multi-sided platform is both a platform and a market intermediary (Hagiu, 2007), thus distinct groups of customers and complementors interact through multi-sided platforms (Boudreau and Hagiu, 2009).

Multi-sided platforms can increase the social surplus for its participants if three conditions are met: (1) there are distinct groups of customers; (2) a member of one group benefits from having his demand coordinated with one or more members of another group; and (3) an intermediary can facilitate that coordination more efficiently than bi-lateral relationships between the members of the group (Evans, 2003; Rochet and Tirole, 2002; Rochet and Tirole, 2003; Rochet, 2003; Armstrong, 2002; Parker and Van Alstyne, 2002).

### 2.1.3 Open-Source Based Platforms

Platforms may be proprietary or open depending on the availability complementarities have on the core component. In a proprietary platform, a single firm serves as platform provider for all platform users (e.g., eBay, Sony Playstation) (Van Alstyne et al, 2006), firms in this position have a strong bargaining power over the rest of the components and are in position to exercise platform leadership. The platform provider often has erected a wall of property rights, therefore effectively controls a ‘bottleneck’ essential to other players (Rochet and Tirole, 2004; Jacobides et al., 2006). According to Boudreau and Hagiu (2009) platform regu-
lators may also have access to a wider menu of regulatory instruments to implement desired actions. Apart from licensing, property rights assignment and other traditional contractual and legal instruments, platform technologies and design are themselves understood as a means of virtually imposing laws (Lessig, 1999) and design rules (Baldwin and Clark, 2000) by the platform leaders.

Flexible technologies on the one hand, and learning imperatives on the other, are now forcing organizations to open their “cores” by placing them in the middle of networks of relations with other organizations (Lomi, 1997). Eisenmann et al (2009) argued platform is open to the extent that: (1) restrictions are not place on participation in its development, commercialization or use; and (2) any restrictions - for example, requirements to conform with technical standards or pay licensing fees- are reasonable and non-discriminatory, that is they are applied uniformly to all potential platform participants.

The goal for firms in an open-source based platform is to reap benefits from the contribution of a group of firms who prefer a loose relationship (Goldman and Gabriel, 2005). In software development, opening the platform requires the use of an open source, which is available to anyone for modifications and the distribution of this improved version (Goldman and Gabriel, 2005). Dedrick and West (2003) argued that in open source based platforms the core element is open without restrictions to any firm willing to participate in the platform; it can alter it to its benefit.

In a series of studies on operating systems, Evans et al (2006) noted that open source is based on a decentralized method for producing software that relies heavily on the Internet. Programmers working on their own or through their companies contribute code to open-source projects. The source code of the resulting programs is made available for free; hence the term open source. Not only stand-alone programmers work in open-source based platforms, many of the contributors are employed and paid by large corporations. Is based on this openness to contribute that open-source based platforms have a high potential for creating value, Chesbrough and Appleyard (2007) mentioned that firms that open their innovation process will benefit from a pool of knowledge and low prices due to the use of open sources.
Linux operating system is open in all its different actors, while Windos OS on the other hand appropriates value by closing its architecture in the design of the platform and retaining the IP rights, therefore the strategy the company follows is closing the platform sponsor level but leaving open the rest of the roles so more complementors feel attracted and interact with the same platform (Eisenmann, 2009). A platform that has been quite closed in its platform architecture is the iPhone by closing most of its roles and leaving only the end user role to decide its final configuration of the product, not from outsider but from Apple since it has closed all the architecture (See table 1).

<table>
<thead>
<tr>
<th>Role</th>
<th>Linux</th>
<th>Windows</th>
<th>Macintosh</th>
<th>iPhone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Demand-side user (End user)</td>
<td>Open</td>
<td>Open</td>
<td>Open</td>
<td>Open</td>
</tr>
<tr>
<td>Supply-side User (Application developer)</td>
<td>Open</td>
<td>Open</td>
<td>Open</td>
<td>Closed</td>
</tr>
<tr>
<td>Platform provider (Hardware OS bundle)</td>
<td>Open</td>
<td>Open</td>
<td>Closed</td>
<td>Closed</td>
</tr>
<tr>
<td>Platform sponsor (Design and IP rights owner)</td>
<td>Open</td>
<td>Closed</td>
<td>Closed</td>
<td>Closed</td>
</tr>
</tbody>
</table>

**Table 1:** Comparison of openness by role in platforms (Adapted from Eisenmann, 2009)

Studies in modelling have indicated that open technologies might be improved faster than protected technologies suggesting that in some situations open technology may outcompete protected technologies (Schilling, 2009; Dalle and Jullien, 2003; Casadesus-Masanell and Ghemawat, 2003). Another benefits from adopting or participating in an open-source based platform range from a sustained stream of producer investment in the technology and access to a large supply of complementary assets (Dedrick and West, 2003).

However the openness of open-source based platforms stimulates its complexity, which in turn increases the network effects. These effects lead a single change in a module to affect the entire system (Gawer and Cusumano, 2002). Moreover innovation processes within open-source based platforms rise from the recognition of an opportunity in a creative and open-minded working environment where participants in the process are able to transcend their personal as well as organizational boundaries (Leonard and Sensiper, 1998; O’Connor
Coordination Dynamics in Open-Source Based Platforms

and McDermott, 2004). And it is this transcendental way of working that creates an scenario where there are no clear ties, obligations or coordinative mechanisms between the participants.

Therefore open-source based platforms challenges the synchronization of innovation in the system as a whole, as Adner and Kapoor (2006) argued focal firms within platforms may be dependent on external actors in its ecosystem overcoming their own innovation challenges and the existence of ecosystem bottlenecks limiting the innovation process within networked firms. In such scenario innovation success depends not only on a firm’s own successful completion but on the successful development and deployment of all other components of the system. And the more dependent an innovation is on other developments, the less control it has over its own success (Adner, 2006).

This complexity limits the communication and coordination in platforms as Goldman and Gabriel (2005) argued the most critical problems for open-source strategies is the duplication of efforts in developing a module of the platform, the control efforts by groups of interest within the platform and most importantly the belief that the community will arise by itself. Therefore stand alone strategies don’t work when a company depends on the collective health of the organizations that influence the creation of a product (Lansiti and Levien, 2004).

2.1.4 Platform Leadership

Industry platforms need to be synchronize for the market to work well for all of its actors, therefore some companies have taken a more dynamic role in the platform and influenced others to pursue a “common” vision of the platform. Jarillo (1988) argued the existence of a “hub firm”, which is the firm that set up the network, and takes a proactive attitude and the care of it. The author argues without some sort of governance, a group risks becoming no more than a haphazard collection of alliances.

However Gawer and Cusumano (2002) first described the leadership role of firms within platforms, according to the authors, platform leaders are companies that drive industrywide innovation for an evolving system of separately developed pieces of technology. In the field of systems Prencipe (2003) argued that networks form of organizations are led by firms that
Coordination Dynamics in Open-Source Based Platforms

manage external relationships, enable firms to exploit variety and have authority to deal with and implement changes. According to Boudreau and Hagiu (2009) platform leaders are found at the core of the technical design defining system architecture and technical relationships.

In their book on platform leadership, Gawer and Cusumano, (2002) presented the example of how Intel drove innovation across the PC industry platform when the platform was fragmented and the platform was constraining the development of Intel’s microprocessors. Intel sponsored innovation, stimulated external innovations on components and coordinated industrial innovation across boundaries to influence industry innovation and drive platform evolution. However Intel’s negotiation capabilities were strongly supported by its bargaining power and brand recognition.

The platform leader process of “negotiation” is present in several industries, Evans et al (2006) discussed how in the PC platform independent hardware producer need to be courted by the software platform provider. Microsoft needs to convince computer makers to build computers that run on Windows. PalmSource must convince companies to design and produce PDAs based on the Palm OS. This courtship takes the form of a rather intricate dance in periods of rapid innovation and changing technical standards, since quick changes in hardware and software need to be coordinated among independent firms if systems are to function well and the ecosystem as a whole is to prosper.

Platform leadership nature resides in the fact that a platform leader has a strong bargaining power over the rest of the participants and therefore can influence the innovation path of the industry platform. Platform leaders’ bargaining power is enhanced in closed platforms where the leader often has erected a wall of property rights. Therefore the platform owner effectively controls a ‘bottleneck’ essential to other players (Rochet and Tirole, 2004; Jacobides et al., 2006). According to Boudreau and Hagiu (2009) platform regulators may also have access to a wider menu of regulatory instruments to implement desired actions. Apart from licensing, property rights assignment and other traditional contractual and legal instruments, platform technologies and design are themselves understood as a means of virtually imposing laws (Lessig, 1999) and design rules (Baldwin and Clark, 2000) by the platform leaders.
Another problem for platform leadership is that the boundary between platform and complementary solutions are not well defined, and platform owners constantly absorb innovative features of complementary applications into the platform (Gawer and Henderson 2007). In doing this platform leaders may discourage innovation among complementors.

Coordination under the umbrella of platform leadership it may become a long and expensive process. This was the case for Intel where it took five years and billions of dollars for the company to convince complementors to increase its capabilities in order to match Intel’s microprocessor full potential (Gawer and Cusumano, 2002). In terms of innovation this approach presents a strong deficiency: the direction of the innovation path might respond to the platform leader needs and its expectations of the technology not of the whole platform. As Lansiti and Levien (2004) mentioned, keystone firms do not promote the health of others for altruistic reasons but because it is beneficial for their own strategy.

In open platforms where bargaining power is not based on property rights and where everybody has access to the same core platforms, platform leadership is often diffuse, while Gawer and Cusumano (2002) argued that open platforms do not have leaders, Evans et al (2006) argued that when several ecosystem participants have critical knowledge, leadership in the innovation process is often shared. And the identity of the leaders is not necessarily predetermined. These relationships involve regular exchange of information and joint work on defining new standards and specifications. In another perspective Boudreau and Hagiu argues that the role the platform itself plays an important role in regulating the surrounding ecosystem.

Coordinating innovation particularly in open source platforms is difficult to achieve by a single firm, hence the need to coordinate efforts across the platform. Regarding this issue Evans (2006) argued that even though software platforms must rely on third parties: increasing technological complexity and consumer demand for more diverse and better products make it impossible for the same firm to innovate effectively throughout the entire system; enlisting the cooperative participation of outsiders via well-defined interfaces becomes a must.

In addition to these issues it has been documented that authority is not very effective in managing uncertainty, if this last construct includes cognitional complexity and high re-
search and problem-solving intensity (Bavelas 1951; Perrow 1967; Burns and Stalker 1961). Lomi (1997) makes clear that during when product conception (involving for example the design of a complex product or service) and execution (involving for example the actual production or delivery) are separated by clear corporate and institutional boundaries authority loses much of it value as coordination mechanism. Therefore and given that open-source based platforms are developed in a rapid change environment where uncertainty must be tackled by its members, the platform leadership view does not apply for this complex scenario.

2.2 Coordination Theory

According to Bergman et al. (2009) successful innovation requires control in the open innovation paradigm, as firms have to reconfigure structures and processes continuously to match with the changing business environment. Firms need capabilities and well-designed processes to manage knowledge flows and coordinate relationships with their innovation partners. Thus, innovation management also means orchestrating complex social processes in which interaction between different actors creates new knowledge and reveals new business opportunities. Still, there are limits in the effective use of external sources. Moreover Eisenmann (2008) argued shared platforms require governance arrangements that balance the inevitable tension between cooperation and competition. Therefore and given that platform leadership strategy do not by itself explain the development of coordination dynamics in the complex scenario present in open-source based platforms a review of the coordination theory will be carried in the present section to propose coordination structures, processes and mechanisms. In order to be consistent with the open-source based scenario coordination theory used for this section is set up at an interorganizational level.

This section is organized as follows, first coordination definitions are reviewed, then coordination is classified from a structural perspective, following by the identification of the main coordination dynamics or processes used to support the structures and ultimately the coordination mechanisms that support the processes are discussed.

Coordination has been defined as the process of “managing dependencies between activities” (Malone and Crowston, 1994). Singh (1992) defined coordination as “The integration and harmonious adjustment of individual work efforts towards the accomplishment of a lar-
ger goal”. Thompson (1967) argues coordination concerns the combination of parts to achieve the most effective or harmonious results. Curtis (1988) in a business context defined coordination as “Activities required to maintain consistency within a work product or to manage dependencies within the workflow”. Therefore, all activities that involve more than one actor require (1) some way of dividing activities among the different actors and (2) some way of managing the interdependencies among the different activities (March and Simon 1958; Lawrence and Lorsch 1967). Wren in a systems perspective mentioned that firms cooperate to achieve some larger system objective,

Beer, Eisenstat, and Spector (1990) pointed out that coordination is necessary for innovation and competitive success. They claimed that cooperation is a prerequisite of coordination and that motivational factors are in turn prerequisites of cooperation. (Smith et al 1995).

In this sense Van de Ven and Walker (1984) argued the perceived need for resources to achieve organizational goals is clearly the most important factor that stimulates interorganizational coordination. Oliver (1990) argued that the decision to initiate relations with another organization is commonly based on multiple contingencies such as necessity, asymmetry, reciprocity, efficiency, stability, and legitimacy. Axelrod (1984) identified pooling resources and responding to a threat as the two conventional reasons for establishing cooperative efforts. In interorganizational links, coordination is thought to enhance the innovative capabilities of organizations by providing opportunities for shared learning, transfer of technical knowledge, legitimacy, and resource exchange (Nohria & Eccles, 1992). Firms use interorganizational coordination to acquire new technologies and expand their product-market reach (Pennings and Harianto, 1992).

### 2.2.1 Institutionalization of Coordination

The phenomenon covered under the concept "coordination" ranges from simple ad hoc agreements between two organizations to participation in formally organized coordination councils (Wethen, 1981). In addressing this issue Introna and Petrakaki (2007) argued coordination is matter of degree, in this sense most contemporary interorganizational cooperation will present different degrees of coordination ranging from virtual to institutionalized coordination.
Firms within interorganizational relationships have increasingly employed formally structured arrangements for coordination (Van de Ven and Walker, 1984). Ring and Van de Ven (1994) noted that cooperative relationships are "socially contrived mechanisms for collective action, which are continually shaped and restructured by actions and symbolic interpretations of the parties involved". Supporters of institutionalized coordination argue when the interorganizational system is structurally fragmented, coordination is low; when it is structurally integrated, coordination is high (e.g. Bolland and Wilson, 1994). The use of manufactured forms of structures represents a shift from a virtual coordination to the institutionalization of coordination in look for high levels of coordination.

Institutionalization is a socialization process that transforms an instrumental transaction into a socially embedded relationship by infusing it with norms and values that permit the relationship to be reproduced and perpetuated beyond the immediate tenure of its founders (Berger & Luckmann, 1966). Shull, Delbecq, and Cummings (1970) discussed, institutionalization both objectifies and internalizes the identity, mission, and procedures of a relationship into taken-for-granted (congruent) expectations by its participants; it colors all aspects of the relationship and gives it a social integration that goes far beyond its formal, legal structure of governance and economic exchange.

According to Ring and Van de Ven (1994) the institutionalization of a relationship is evident in three basic interactions that evolve over time between formal and informal processes of negotiation, commitment, and execution: (a) personal relationships increasingly supplement formal role relationships, (b) psychological contracts increasingly substitute for formal legal contracts, and (c) as the temporal duration of relationships extend beyond the tenure of initial contracting agents, formal agreements (e.g., rules, policy, contracts) increasingly mirror informal understandings and commitments. Another form of institutionalization of coordination is through the formation of decentralized institutions (Wren, 1967; Whetten, 1981; Rosenkopft and Tushman, 1996; O’Mahoney and Ferraro, 2007).

The formation of a central institution to achieve coordination has taken several forms in different industries, in the aerospace industry, Wren (1967) identified systems’ coordination was achieved by instituting a formal office which has the authority to make subsystem efforts consonant with the total system. It is here that authority can be focused, coordination
Coordination Dynamics in Open-Source Based Platforms

(linkage) obtained, rivalries and split commitments resolved, and information transfer obtained. The author termed these institutions as interfaces, the interface concept was used to illustrate how systems can be studied, linked, and integrated. The interface is created when people, organization, or systems must meet in support of one another. The interface is the contact point between relatively autonomous organizations which are nevertheless interdependent and interacting as they seek to cooperate to achieve some larger system objective.

Rosenkopft and Tushman (1996) found that industries characterized by complex technologies, like flight simulation, rely on cooperative groups such as technical committees, task forces and standards bodies to adjudicate the process of technological evolution and termed them 'cooperative technical organizations' (CTOs). The authors argued CTOs enable exploration of how community-level organization both influences and responds to technological change, as CTOs link various constituencies in pursuit of technological standards and subsequent technological trajectories. Membership of a CTO spans multiple constituencies, such as firms, government and academia. Yet membership in a CTO is not open to all: the leaders of CTOs limit group sizes and select new members.

Industries characterized by a high level of competitiveness and short cycles of innovation put firms into vulnerable strategic positions (Eisenhardt and Schoonhoven, 1996) and generate the need to spread risk and acquire additional resources and competences through the formation of strategic alliances (Rosenkoft and Schleicher, 2008). The nature of strategic alliances can vary widely. For example, firms might attempt to obtain greater efficiencies of scale by pooling resources within common functional areas (such as merging R&D resources), take advantage of complementary skills by pooling resources across functions (such as teaming R&D resources and marketing functions), or develop new products in parallel (Amaldos et al, 2000). According to Whetten (1981) this type of coordination involves the formation of a central administrative unit that responds to the wishes of member agencies and often is associated with a federation.

These forms have been also found in open source based platforms particularly in developers’ communities. In a study on the open source Debian Linux community, O’Mahoney and Ferraro (2007) showed how the community transitioned from a de facto
Coordination Dynamics in Open-Source Based Platforms

governance model to one that integrated positional authority with directly democratic means. During this transition the open source community of developers went through a series of governance phases:

- **De facto governance**: In this phase autocratic leadership emerges and eventually is challenged by the community.
- **Designing governance**: Formal authority is developed and is limited through democratic means.
- **Implementing governance**: Varying conceptions of formal authority are debated and soon after community members elect leaders through democratic means.
- **Stabilizing governance**: A shared conception of formal authority emerges.

The institutions above mentioned were designed to increase the level of coordination among its members. The institution is a response to the problem of knowledge sharing because they concentrate and transfer knowledge to all of the members (see figure 3). In high technology industries this institutions are conceived as accelerators of innovation and ease the adoption of technology standards. The institutionalization of coordination therefore is the creation of an institution responsible for easing the reach of common goals by promoting, enhancing and linking processes within organizations. According to Wren (1967) this types of institution are the supreme coordinating agency.

![Figure 3: Institutionalized Coordination](image)

It is important to mention that the structure creates the context for coordination but it does not represent the process of coordination itself. This process must be induced within the walls of the institution (Whetten, 1981).
Coordination Dynamics in Open-Source Based Platforms

In this section it was discussed coordination is matter of degree, in this sense most contemporary interorganizational cooperation will present different degrees of coordination ranging from virtual to institutionalized coordination. In this sense coordination was categorized as virtual and institutionalized, virtual been the lowest degree of structure and institutional coordination as the most structured form of coordination.

Proposition 1: Firms around an open-source based platform will engage in the creation of an institution responsible for promoting, linking and regulating coordination of all the different actors.

2.2.2 Coordination Processes

The implementation of interorganizational processes among firms to coordinate its efforts has giving rise that it has been termed virtuality (Chebbi et al., 2005). Virtuality is the ability of the organization to consistently obtain and coordinate critical competences through its design of value-adding business processes and governance mechanisms involving external and internal constituency to deliver differential, superior value in the marketplace (Sieber and Griese, 1997).

The strategic aims of virtuality are often reached by establishing a business network which is supported by interorganizational systems (Sieber and Griese, 1997). According to the Sieber and Griese (1997) virtuality main features are the interaction between independent companies to create a common value and distributed, predominantly information-based processes, resulting in location independence. The interorganizational processes, shared goals and agreements made by firms has come to be termed as virtual organization (see figure 4) by several authors (e.g. Chebbi et al. 2005; Dess et al., 1995; Introna and Petrakaki, 2007).

Having set the virtual aspects of coordination through processes, in the following section different coordination processes across interorganizational relationships literature will be discussed in order to identify the common components in coordination processes to draw a generic process proposal.
If coordination is defined as managing dependencies, Malone and Crowston (1994) suggest further progress should be possible by characterizing different kinds of dependencies and identifying the coordination processes that can be used to manage them.

Malone and Crowstone (1990) identified four processes underlying coordination: coordination, group decision making, communication and perception or common objects. (1) At the coordination process level the activities carried are identifying goals, ordering activities, assigning activities to actors, allocating resources and synchronizing activities. (2) In group decision making process actors are involved in proposing alternatives, evaluating alternatives, making choices (e.g., by authority, consensus or voting). (3) The communication process requires establishing common languages, selecting receiver (routing) and transporting message (receiving). (4) Perception of common objects, this process requires the representation of common objects (such as activities and bids) which both senders and receivers can perceive.

However in a subsequent paper on coordination, Malone and Crowstone (1994) expanded the reach of coordination processes to include the management of shared resources, relationship between producer/consumer, simultaneity constraints and task/subtask dependencies.

In the author’s view whenever multiple activities share some limited resource (e.g., money, storage space, or an actor’s time), a resource allocation process is needed to manage the interdependencies among these activities.

The second process Malone and Crowstone (1994 suggest concerns a “producer/ consumer” relationship, that is, a situation where one activity produces something that is used by an-
other activity. Producer/consumer relationships often lead to several kinds of dependencies:

1. Prerequisite constraints: When this dependency exists, there must at least be some notification process to indicate to the consumer activity that it can begin. Managing prerequisite dependencies also often involves explicit sequencing and tracking processes to be sure that producer activities have been completed before their results are needed.

2. Transfer: When one activity produces something that is used by another activity, the thing produced must be transferred from the “producer” activity to the “consumer” activity. Managing this dependency usually involves physical transportation in the case of physical goods however when the thing transferred is information, the transfer is “communication” rather than transportation.

3. Usability: Another, somewhat less obvious, dependency that must often be managed in a producer/consumer relationship is that whatever is produced should be usable by the activity that receives it. One common way of managing this dependency is by standardization, creating uniformly interchangeable outputs in a form that users already expect. This is the approach on assembly lines, for example. Another approach is to ask users what characteristics they want. A third, related, alternative is participatory design, that is, having the users of a product actively participates in its design (Schuler and Namioka 1993).

The third process is managing simultaneity constraints, which refers to dependencies that need to occur at the same time (or cannot occur at the same time). In general, the instructions of the different processes can be executed in any order. Permitting this indeterminacy improves the performance of the system (e.g., one process can be executed while another waits for data to be input) but can cause problems when the processes must share data or resources (Malone and Crowstone, 1994).

The fourth process in this series concerns the dependencies between managing task/sub-task dependencies, according to Malone and Crowston (1994) this dependencies can be managed by two forms:
1. Top-down goal decomposition: The most commonly analyzed case of managing this dependency occurs when an individual or group decides to pursue a goal and then decomposes this goal into activities (or subgoals) which together will achieve the original goal. In this case, we call the process of choosing the goal selection and the process of choosing the activities goal decomposition.

2. Bottom-up goal identification: Another possibility is that several actors realize that the things they are already doing (with small additions) could work together to achieve a new goal. For example, the creation of a new interdisciplinary research group may have this character. In human systems, this “bottom-up” process of goal selection can often engender more commitment from the actors involved than a top-down assignment of responsibility.

In a more systemic view, Zajac and Olsen (1993) proposed a stage model of cooperative relationships composed of an initializing stage, a processing stage, and a reconfiguration stage with feedback loops to the earlier stages. The first stage is an 'initializing' stage, in which each firm formulates its own strategic plans, subjectively evaluates its exchange alternatives, and begins its involvement in interorganizational exchange. This stage also includes the first rounds of exchange. These often take the form of preliminary communication and negotiation concerning mutual and individual firm interests, and/or feasibility studies and general information exchange.

The second stage, called the 'processing' stage, encompasses the forecast period over which value-creating exchanges in the interorganizational strategy are expected to occur. This stage focuses on behaviors associated with processing the formal and informal mutual obligations that create value, as well as the distribution of those gains over multiple rounds. Value is not only created, but also claimed and distributed throughout this processing stage. Explicit or implicit norms for managing the divergence of interest will arise to the extent that these norms will be used as conflict resolution systems and trust.

The third stage in the developmental process of an interorganizational strategy is called the 'reconfiguring' stage. This stage, which represents a potential redefinition of the interorganizational strategy. The reconfiguring stage may not involve a change in the type of inter-
organizational strategy per se, but only a change in the process of interaction within the existing interorganizational strategy (Zajac and Olsen, 1993).

Ring and Van de Ven (1994) proposed a number of factors that allow cooperative relationships to evolve or dissolve over time. Treating cooperation as a dynamic process where participants constantly evaluate their decision to continue to cooperate is useful. The authors argue the development processes associated with cooperative interorganizational relationships are cyclical, not sequential. The process is divided in a negotiation stage, commitment stage and in the execution stage.

In the negotiations stage, the parties develop joint (not individual) expectations about their motivations, possible investments, and perceived uncertainties of a business deal that they are exploring to undertake jointly. In this stage the focus is on the formal bargaining processes and choice behavior of parties as they select, approach, or avoid alternative parties and as they persuade, argue, and haggle over possible terms and procedures of a potential relationship. Underlying these formal bargaining proceedings are social-psychological processes of sense making, or enactment, that lead otherwise independent parties to enter into negotiations with one another. (Ring and Van de Ven, 1994)

In the commitments stage, firms reach an agreement on the obligations and rules for future action in the relationship. At this point, the terms and governance structure of the relationship are established, and they are either codified in a formal relational contract or informally understood in a psychological contract among the parties. Depending on the degree of business risk and the willingness of the parties to rely on trust, many of these commitments will be reached informally with a handshake. However, legal agents of the parties often are called upon to formally draft a legal agreement on other key commitments in order to avoid legal impediments. (Ring and Van de Ven, 1994)

Finally, in the executions stage, the commitments and rules of action are carried into effect; formally designated role behavior by the parties reduces uncertainty when they execute commitments, and it makes interactions among parties predictable. Through a series of role interactions, parties also may become more familiar with one another as persons, and they may increasingly begin to rely on interpersonal, as opposed to inter-role, relationships. (Ring and Van de Ven, 1994)
Coordination Dynamics in Open-Source Based Platforms

With time, misunderstandings, conflicts, and changing expectations among the parties are inevitable, and these factors can provide cause for rethinking the terms of the relationship. In these renegotiations, new supplemental agreements typically are established to resolve only the contested issues, but all other terms and understandings contained in the relational contract remain in effect. In this way, the ongoing relationship is preserved. (Ring and Van de Ven, 1994).

Das and Teng (1998) presented a process in which a balance between trust and control mechanisms are set to achieve coordination between firms. That is, the trust level and control level jointly and independently contribute to the level of partner cooperation. The logic of using control is that, through the establishment of proper control mechanisms, the attainment of desirable goals becomes more predictable. Indeed, the purpose of control is to fashion activities in accordance with expectations so that the ultimate goals of the organization can be attained. Just as control mechanisms are meant to enhance the probability of having the desired behavior, trust is also useful in enhancing the perceived probability of desired behavior.

In a sequential view Martinez and Jarillo (1989) argued the process of coordination should include (1) Shaping a formal structure, (2) centralization of authority, (3) formalization and standardization: written policies, rules, job descriptions, and standard procedures. (4) Planning: strategic planning, budgeting, functional plans, and scheduling. (5) Output and behavior control: reports and direct supervision.

The coordination processes presented above lead us to identify several commonalities and devise a generic coordination process. The first stage is an initial planning stage in which firms engage in the generation of a strategic common goal by proposing evaluating and making choices by general consensus. This is followed by the second stage where the delegation of responsibilities across firm boundaries takes place, development of procedures and division and subdivision of tasks. The third stage relates to execution (where the value is created), in this stage firms engage in processes of execution to attain common goals, they do this by using common language, dependency and simultaneity dependencies across activities. The fourth process can be depicted as output control it is here that processes and behavior are controlled by rules, procedures and coordination mechanisms created for the
firms and supervised by the firms or the coordination institution. The final stage is the re-
configuration of strategy by the members once the execution of activities has reached the
established goal.

Proposition 2: Firms arrayed in an open-source based platform will engage
which each other in a system of coordination processes that
will include planning, delegation, execution, control and re-
configuration stages.

A interorganizational relationship that it is strongly binded through processes it is said to be a
virtual organization and It will have neither central offices nor organizational charts, nor hi-
erarchies, and no vertical integration (Byrne,1993). Chebbi et al. (2005) argued virtual or-
ganizations are distributed in time and in space sharing resources and competencies (similar
or dissimilar) and cooperating to reach some shared objectives using information technolo-
gies. The benefit is that, just at virtual memory increases storage capacity, the virtual or-
ganizations enhance the capacity or competitive advantage of participating firms (Dess et
al., 1995). Each company that links up with others to create a virtual organization contrib-
utes only what it considers its core competencies.

Virtual organizations depend upon team working, i.e. virtual teams. Virtual teams are tem-
porary groups of people with very often different cultural and historical backgrounds, who
are geographically dispersed, work with autonomy and interact through ICTs (Jarvenpaa and
Leidner, 1999; Yoo and Alavi, 2004). For a virtual organization to be effective such virtual
teams should be able to establish commonly shared work practices, language and norms.
Thus, virtual teams must not only develop capabilities to work (often in stressful situations)
with electronic information and evolving communication technologies, but they must also
work with a variety of partners who have their own competence and who are leaders in
their own right (Introna and Petrakaki, 2007).

Given that firms linked through coordination processes must carry specific tasks they
strongly rely in virtual groups of people designated to carry out these projects. The creation
of these groups is the response to the need to share core competencies to deliver value to
the firms engaged in interactions.
Proposition 3: Firms in open-source based platforms will create virtual groups to carry interorganizational tasks related to the attainment of common goals of the interorganizational relationship.

2.2.3 Coordination Mechanisms

Coordination processes in order to be implemented successfully actors must perform additional tasks, which Malone and Crowston (1994) called coordination mechanisms. Coordination mechanism is any administrative tool for achieving integration among different units (Martinez and Jarillo, 1989).

Mechanisms of coordination could be divided roughly into two groups: structural or formal and informal mechanisms (Martinez and Jarillo, 1989). Van de Ven et al (1976) classified interorganizational coordination in formal and informal, the author termed the formal coordination as “programming “ and the informal as “feedback. Moreover, in a study on coordination in forty five articles on the subject conducted by Smith et al. (1995), found the literature classifies coordination either in formal or informal, either by examining informal as opposed to formal aspects of cooperative relationships, or transitions from formal relationships to informal ones.

Farrel and Saloner (1988) in a study of standard committees found three common mechanisms for achieving coordination. The first involves explicit communication and negotiation before irrevocable choices are made. The second mechanism, by contrast, involves no explicit communication and depends on unilateral irrevocable choices. It succeeds if one agent chooses first and the others follow. Third, the authors examined a hybrid of the first two mechanisms, in which both communication and unilateral preemptive actions are allowed. This means that firms participating in interorganizational relationships can coordinate using formal, informal or a mix of both forms of coordination mechanisms. (Grandori and Soda, 1995; Van de Ven, 1976; Smith et al., 1995). Also it has been argued by Ring and Van de Ven (1994) that formal types of cooperation can evolve over time into informal types in which rules and regulations are no longer needed.

Informal coordination is a voluntary cooperation and self-coordination by the elements themselves (Wren, 1967). According to Smith et al. (1995) informal cooperation involves
adaptable arrangements in which behavioral norms rather than contractual obligations determine the contributions of parties. Axelrod (1984) discussed the conditions under which such cooperation spontaneously arises; these conditions include the parties’ perceiving they will be in contact with each other for a long time, their believing it is to their advantage to cooperate, and their recognizing they must reciprocate for any benefits received, employing a tit for tat strategy. Astley (1984) referred to this type of cooperation as voluntaristic and organic.

According to Whetten (1981) successful voluntary coordination depends first in perceptual assessments, in other words the positive attitude towards coordination, the recognition of needs for coordination, knowledge of potential partners and assessment of compatibility and desirability. And second to resource and structural adequacy which is the capacity for maintaining coordination linkage.

According to Martinez and Jarillo (1989) informal coordination mechanisms can be divided in (1) lateral relations, (2) informal communication: personal contacts among managers and meetings. (3) Socialization: building shared strategic objectives. As it is noted coordination mechanisms have the common objective, to ease communication the different actors.

Trust often is argued to be the principal mode of social control in interorganizational relationships (Adler, 2001; Ring and Van de Ven, 1992). Additionally Dekker (2004) argues trust as the immediate antecedent of informal cooperation and coordination. Ring and Van de Ven (1994) defined trust as an individual’s confidence in the good will of the others in a given group and belief that the others will make efforts consistent with the group’s goals. Rousseau, Sitkin, Burt, and Camerer (1998) define trust as “a psychological state comprising the intention to accept vulnerability based upon positive expectations of the intentions or behavior of another”.

According to Dekker (2004) Mechanisms to build trust are deliberate risk taking and increasing interaction, for instance by joint goal setting, problem solving, decision making and partner development activities (Das & Teng, 1998; Kale et al., 2000; Saxton, 1997; Uzzi, 1997). Another way of deliberately generating trust and thereby mitigating control problems before a governance structure is designed and implemented is selecting an appropriate partner, based on good predictors of desirable cooperative behaviors (Grandori & Soda, 1995;
Ireland et al., 2002; Jones et al., 1997; Ouchi, 1979). In addition to these mechanisms Das and Teng (1998) added equity preservation, communication and interfirm adaptation as the main mechanisms for trust creation.

According to Van de Ven et al. (1976) feedback is another important source of informal coordination and is based on social interactions and is vested in a group of role occupants through scheduled or unscheduled staff or committee meetings. Thompson (1967) defined coordination by feedback as mutual adjustments made by the members of a group of firms based upon new information.

According to Das and Teng (1998) socialization provides much needed interaction among managers from the firms, thus enabling managers to familiarize themselves with their partner’s organizational culture. Such personal interaction helps to the development of common values and norms for the alliance.

According to Dess et al. (1995) information technology play a major role in coordinating efforts and it should be part of the coordination strategy. Information technology reduces coordination costs and encourages a shift toward the use of more “coordination-intensive” structures. In other words, coordination structures that were previously too “expensive” will now become more feasible and desirable (Malone and Crowston, 1994).

Formal coordination is a mean to control outcomes to be realized by the organizations within an interorganizational relationship and monitor the achievement of these performance targets (Dekker, 2004). In order to attain those targets firms establish formal coordination mechanisms that include hierarchical arrangements (Wren, 1967) that will facilitate this process. Das and Teng (1998) argued formal coordination tends to be especially relevant in strategic alliances, owing to the relatively high degrees of goal incongruence and performance ambiguity in such scenario.

Formal coordination and control is intended to specify how firms within interorganizational relationships should act and monitor whether actual behaviors comply with this pre-specified behavior (Dekker, 2004). Das and Teng (1998) suggest that behavior monitoring consists of elements such as reporting and checking devices, written notice of any departure
from the agreement, accounting examination, cost control, quality control, arbitration clauses, and lawsuit provisions.

Dekker (2004) divided coordination mechanisms in “ex-ante” and “expost” mechanisms. Ex-ante control mechanisms mitigate control problems by aligning partners’ interests and by reducing coordination needs before implementing interorganizational relationships. Within this type of coordination mechanisms are pre-established plans, schedules, forecasts, formalized rules, standard procedures and policies (Gulati and Singh, 1998; Van de Ven, 1976; Wren 1967). The common element of each of these exemplary mechanisms is that a codified blueprint of action is impersonally specified (Van de Ven, 1976).

In Strategic alliances’ ex-ante coordination is characterized by contractual agreements and formal structures of control (Smith, 1995). However, Das and Teng (1998) identified goal setting, structural specifications as the heart of formal control in strategic alliances. Routines, rules and procedures, are used in horizontal clusters (Brusco 1982). Planning, programming and information systems were identified as coordination mechanisms in interfirm applications of information technology (Rockart and Short, 1989). Various forms and degrees of property-right sharing mechanisms, giving rise to proprietary networks were found in joint ventures, cross-partnerships and capital ventures (Daems, 1983). The development phase of these types of mechanisms will require and intense participation of the firms involved and due to its impersonal characteristics, the interaction among firms will decrease after their development.

In the case that “ex-ante” formal control mechanisms do not solve coordination problems are managed by “expost” control mechanisms that achieve control by processing information and evaluating performance (Ouchi, 1979). Some of the mechanisms employed are authority, dispute resolution, standardized information and communication systems (Van de Ven, 1976; Wren, 1967). In strategic alliances “expost” mechanisms are exercised in the form of decision making meetings among the members (Das and Teng, 1998). Group problem solving supported by committees, task forces, panels and user groups has been found in joint R&D (Schrader, 199). However, interfirm authority was identified in clusters of local firms accepting the leadership of a single firm as expost coordination mechanism (Lorenzoni and Ornati, 1998). Given the characteristics of expost control mechanisms, the interaction
of firms required for their application will be more intense than in the “ex-ante” mechanisms.

Proposition 4: Firms in the open-based platform will use a hybrid set of coordination mechanisms conformed of informal and formal mechanisms.
3 Methodology

3.1 Research settings

Theoretical sampling (Glaser & Strauss, 1999; Strauss & Corbin, 1990) was used to select an open source based platform that had three features that would allow us to explore interorganizational coordination in depth: an open-source based platform that presents complex composition in terms of number of participants, second, has developed a structure to coordinate its members and third, leadership turnover. Successful leadership turnover implies that leadership positions have become institutionalized or decoupled from the founder of a project. (O’Mahoney and Ferraro, 2007).

The Symbian OS platform has been selected as unit of analysis. The Symbian OS platform has the highest market share in the mobile industry (Canalys 2010), its members have created the Symbian Foundation in an effort to coordinate innovation, currently the foundation has 184 members participating in the platform. Given this facts, the Symbian OS platform has been selected as unit of analysis, specifically the Symbian Foundation will allow this study to examine coordination efforts and links in one of the most complex platforms of the mobile industry.

3.2 Research methodology

Saunders et al. (2007) defined research as, “something that is undertaken by the people to find out the things in a systemic way to increase their knowledge”. The aims and objectives of any research project are largely determined by how much is already known about the topic selected (Shaw, 1999). Therefore it is fairly important to achieve valid and reliable findings for appropriate research methodology. Consequently, the methodology employed here to gather this information is important in the overall research process.

There is limited research available in area of coordination within open source based platforms, therefore, the aim and objectives of this research project is to understand dynamics of coordination within the technology platforms. There can be different research approaches adopted, Saunders et. al. (2003) proposed a research termed the “research process onion”, which provides systematic way through research process, or what Shaw (1999)
named the process of inductive analysis through collecting theoretical and primary/secondary data and conducting analysis to draw up conclusion. However the research process onion by Saunders et al. (2003) has been adopted in this study (see figure 5), because it provides an opportunity to conduct research in an adaptive way related to benefits. The research process onion presents a structural logical sequence; it starts by defining the research philosophy, followed by research approach and strategy.

Figure 5: The research process onion (Saunders et al. 2003)

3.3 Research Approach

Every research project uses the involvement of theory, which in turn may or may not be explicit in presentation of findings and conclusion (Saunders et al. 2003). The clarity about the theoretical framework helps to develop relevant questions about the subject of study. This leads researchers to adopt a deductive or an inductive approach.

The inductive approach is a systematic procedure for analyzing qualitative data in which the analysis is likely to be guided by specific evaluation objectives, whereas, the deductive analysis refers to data analyses that set out to test whether data are consistent with prior assumptions, theories, or hypotheses identified or constructed by an investigator (Thomas, 1996). In practice, many evaluation projects use both inductive and deductive analysis. The approach for this research devised in order to view the relationship between theory and re-
search (Bryman and Bell, 2007) therefore, a framework for theory underlying coordination and platforms is developed to test this theory over empirical information. Consequently, the purpose of this study is to understand how coordination is achieved within open-source based platforms and it is appropriate to adopt the deductive approach rather than the inductive (Saunders et al., 2003). The use of the deductive approach will allow propose a model for coordination within open-source based platforms and then subject the model proposed to empirical scrutiny (Bryman and Bell, 2007).

3.4 Research Strategy

According to Bryman and Bell (2003) a research strategy, is a general orientation to the conduct of business research. The research of a given subject leads to the selection of either qualitative or quantitative research strategies. According to Denzin & Lincoln (1994) the qualitative methods are more natural to interpret the phenomena with detail which is near to human interpretations. Given these developments and because this study deals with the interpretation of the coordination phenomena in open-source based platforms a qualitative method is adopted in this research. A qualitative research suits to the discussion of sensitizing concepts (Bryman and Bell, 2007) used in the development of this study.

The empirical study is based on an extensive examination of a single entity, therefore the research will be a case study (Bryman and Bell, 2007). The use of a case study will allow deepening into the activities within the Symbian Foundation to better understand its dynamics. Saunders et al. (2003) argued that most research projects for academic courses are necessarily time constrained. A cross sectional case study was selected to analyse the Symbian Foundation at a point in time but with retrospective looks to their development in time.

3.5 Data collection methods

From an interpretive perspective, as Eisenhardt (1989) recommends the present study begins with the set of a broad research question, then with the establishment of systematic data collection and developed a case study to create strong triangulated measures. There-
fore observation with interviews and documentary sources were carried to complement these measures (Hammersley and Atkinson, 1983).

Semi-structured interviews were selected as source of primary data, by doing so a list of themes and derivative ideas were covered during the interviews. According to Saunders et al (2003) in semi-structured interviews the order of the questions may be varied depending on the flow of the conversation and additional questions may be covered to explore the events. The nature of semi-structured questions allowed to cover the initial issues of coordination dynamics in open-source based platforms and also to explore further issues not contemplated but relevant to explain the phenomenon. Interviews initially were carried personally, followed by follow ups by phone and by email.

Symbian’s Personnel Interviewed:
- Lauren Sarno, Head of Membership.
- Laura Chuck, HR Business Partner.
- Sebastian Brannstrom, Release Manager.
- Victor Tuson Palau, Delivery Management.
- Scott Weiss, Head of User Experience Council.
- Daniel Rubio Head of the Architecture Council
- Chris Davidson, Programme Manager.

Another source of primary data were “Forums” in which a series of issues regarding coordination within open-source based platforms were posted on a online forum designed for these purposes. In these forums members of the Symbian Foundation discussed through the posting of threads. The benefits of doing so were several: it brought together specialists to discuss the threads posted and enrich the previous answers given by participants. It helped to gather information from different sources, such as videos, books and articles.

This network effect within the forums represented a rich source of information concentrated in one single place, the diversity of the participants lead to non explored perspectives of the issue, it allowed comparisons from the strategy level and the technical level of coordination within open-source based platforms and comparison from the company and the individual point of view as well.
Forum Participants:

- Ian Hutton, Head of Feature and Road Map Council.
- Sebastian Brannstrom, Release Manager.
- Lauren Sarno, Head of Membership.
- Roelof Kotze, Member and Community Support

Symbain foundation’s websites are a rich source of secondary data, from these websites it has been extracted information contained in books, videos, articles, news, and public information. This information is given in detail in the empirical chapter that follows.
4 Empirical Study

Given that the creation and operation of the Symbien foundation is a reflection of the interest of the Symbian based platform’s members to coordinate and to interact with each other, the observation and study of the foundation will shed light over the understanding of coordination dynamics within open-source based platforms. In the following section, the information obtained from the data collection process on the foundation’s composition, processes, interactions and mechanisms is presented.

4.1 History of Symbian

From its earliest days, the idea that became Symbian was all about collaboration—starting with David Potter's early 1980s designs of games and office productivity software for Sinclair's personal computers, a partnership that launched the "Psion" name. Those programs helped give birth in 1984 to the Psion Organizer, the world's first handheld computer—and one that would quickly support a simple-to-use database programming language, OPL (www.symbian.org).

The collaborative support from the industry for the growing power of the Psion software base led to the historic formation in 1998 of Symbian, a joint venture between Psion and phone manufacturers Ericsson, Motorola, and Nokia. Over the next few years Symbian helped bring forth the explosion of mobile device innovation—with Symbian software at the base of more than 100 million phones by 2006.

4.1.1 Early Alliances

In April 1999, Symbian entered into an alliance with Sun Microsystems to allow for the development of applications in Java, a programming language created by Sun that permits compatibility with many other digital products. Motorola, one of its partner companies, entered into an agreement with Cisco for networking applications. Moreover, Motorola's links
with Netscape were tied to this alliance and to AOL and Sun. In October 1999, Nokia and 3Com (the then producer of the Palm Pilot, the most popular PDA in the world), signed an agreement that would allow EPOC to be used as a standard operating system for new devices and the interface of Palm, commonly used in mobile telephones, and the optical pen. During late 1999, Symbian entered into an agreement with NTT DoCoMo, the Japanese company that launched the successful I-mode mobile phone service (Ratliff, 2002). In February 2000, Symbian joined hands with IBM with to jointly develop PDAs based on EPOC software. In the same period, Ericsson, IBM, Lotus, Oracle, Palm and Symbian founded the GPRS Applications Alliance, an alliance devoted to improving the development and diffusion of the GPRS. In April 2000, Symbian formed an alliance with Sony for using EPOC on all Sony PDAs. In August 2001 it linked up with Intel to integrate Epoc with Intel’s chip technology (Ancarani and Shankar, 2003).

4.1.2 Platform Formation

Daniel Rubio, Chief Architect, Symbian Foundation (www.developer.symbian.org):

“In the past it was a company that was offering services to different OEMs, in that respect there was no communication within the different OEMs so it was sort of a hub and the information flooded. We were guessing what the rest was doing, we were small players”

On June 2004, Nokia, Sony Ericsson, Motorola and NTT DOCOMO announced that their intended to unite Symbian OS™, S60, UIQ and MOAP(S) to create one open mobile software platform. Together with AT&T, LG Electronics, Samsung Electronics, STMicroelectronics, Texas Instruments and Vodafone they established the Symbian foundation which was open to all organizations. These companies also received wide support from other industry leaders for what it was a promising step in the creation of a more interlinked platform.

In 2008, the next step of Symbian evolution took place, with Nokia purchasing all Symbian assets and starting the software down the path to open source. As the Symbian Foundation and all its members look to the future and the billions of forthcoming interconnected mobile
devices, it will be innovative collaboration—working together—that will help make people more productive, more creative and more entertained than ever before.

Sony Ericsson, DOCOMO and Motorola contributed with assets to provide a unified platform with common UI framework. The Foundation’s platform was intended to be built on the leading open mobile software platform, with more than 200 million phones, across 235 models, already shipped by multiple vendors and tens of thousands of third-party applications already available for Symbian OS-based devices.

Following are comments by the Symbian’s founding members extracted from the Foundation’s website (www.symbian.org), comments are in relation to the launch of the foundation:

Nigel Clifford, CEO of Symbian:
“Our vision is to become the most widely used software platform on the planet and indeed today Symbian OS leads its market by any measure. Today’s announcement is a bold new step to achieve that vision by embracing a complete and proven platform, offered in an open way, designed to stimulate innovation which is at the heart of everything we do.”

Olli-Pekka Kallasvuo, CEO of Nokia:
“Establishing the Foundation is one of the biggest contributions to an open community ever made. Nokia is a strong supporter of open platforms and technologies as they give the freedom to build, maintain and evolve applications and services across device segments and offer by far the largest ecosystem, enabling rapid innovation. Today’s announcement is a major milestone in our devices software strategy.”

Dick Komiyama, President of Sony Ericsson:
“Sony Ericsson believes that the unified Symbian Foundation platform will greatly simplify the world for handset manufacturers, operators and devel-
opers, enabling greater innovation in services and applications to the benefit of consumers everywhere."

Kris Rinne, Senior Vice President of Architecture and Planning at AT&T:
"Mobile phones have turned into sophisticated multimedia computers and smart phones continue to grow in popularity," "The Symbian Foundation will reduce fragmentation in the industry and holds the promise of incorporating leading technology and the most mature software into a unified platform for the entire industry. This will create an environment that will encourage and enable developers to build compelling applications that will positively affect our customers' lives and support AT&T in offering its differentiated services to consumers."

On the 4th of February 2010 the Symbian Foundation completed the open source release of the source code for the world’s most widely-used smartphone platform. The Symbian platform, which has been developed over more than 10 years and has shipped in more than 330 million devices around the world, is now completely open and the source code is available for free. The transition of this market-leading platform from proprietary code to open source is the largest in software history. The move has been completed four months ahead of schedule and provides the basis for unlimited mobile development based on innovation and openness.

Any individual or organization can now take, use and modify the code for any purpose, whether that be for a mobile device or for something else entirely. This strategic move provides the Symbian ecosystem with greater potential for innovation, faster time-to-market and the opportunity to develop on the platform for free. Symbian’s commitment to openness also includes complete transparency in future plans, including the publication of the platform roadmap and planned features up to and including 2011. Anyone can now influence the roadmap and contribute new features.

Lee Williams, Executive Director of the Symbian Foundation (www.symbian.developer.org):
“The development community is now empowered to shape the future of the mobile industry, and rapid innovation on a global scale will be the result. When the Symbian Foundation was created, we set the target of completing the open source release of the platform by mid-2010 and it’s because of the extraordinary commitment and dedication from our staff and our member companies that we’ve reached it well ahead of schedule.”

All 108 packages containing the source code of the Symbian platform can now be downloaded from Symbian’s developer web site, under the terms of the Eclipse Public License and other open source licenses. Also available for download are the complete development kits for creating applications (the Symbian Developer Kit) and mobile devices (the Product Development Kit).

**Key Dates on Symbian History** (www.symbian.org):

- 1980: Psion founded by David Potter
- 1984: Psion Organiser launched
- 1986: the "vastly improved" Psion Organiser II launches, with a simple-to-use database programming language, OPL.
- 1987: Psion begins development of its "SiBO" ("Sixteen Bit Organiser") family of devices and its own new multitasking operating system called EPOC to run its PDA products.
- 1989: First EPOC16 devices, the MC400 and MC200, ship with a primarily 1-bit, keyboard-operated graphical interface.
- 1997: The first version of EPOC32 Release 1 appeared on the Psion Series 5 ROM v1.0. The EPOC32 operating system, at the time simply referred to as EPOC, was later renamed Symbian OS. EPOC32 was a pre-emptive multitasking, single user operating system with memory protection, which encourages the application developer to separate their program into an engine and an interface.
- 1998: In June Psion Software became Symbian, a major joint venture between Psion and phone manufacturers Ericsson, Motorola, and Nokia. As of Release 6, EPOC became known simply as Symbian OS.
1999: The Psion Series 5mx, Psion Series 7, Psion Revo, Diamond Mako, Psion net-Book, netPad, GeoFox One, and Ericsson MC218 were released using ER5. A phone project was announced at CeBIT, the Phillips Illium/Accent, but did not achieve a commercial release.

2000: The first phone, the Ericsson R380 was released using ER5u in November.

2001: The first 'open' Symbian OS phone, the Nokia 9210 Communicator, was released in June 2001. Bluetooth support was added. Almost 500,000 Symbian phones were shipped in 2001, rising to 2.1 million the following year.

2003: First shipment of Symbian OS 7.0 and 7.0s, an important Symbian release which appeared with all contemporary user interfaces including UIQ (Sony Ericsson P800, P900, P910, Motorola A925, A1000), Series 80 (Nokia 9300, 9500), Series 90 (Nokia 7710), Series 60 (Nokia 3230, 6260, 6600, 6670, 7610) as well as several FOMA phones in Japan. It also added EDGE support and IPv6. One million Symbian phones were shipped in Q1 2003, with the rate increasing to one million a month by the end of 2003.

2004: Psion sells its stake in Symbian.

2006: 100 millionth phone with Symbian OS is shipped.

2008: Symbian acquired by Nokia; Symbian Foundation formed.

### 4.2 Organization

Symbian Foundation Ltd. is a non-profit corporation, incorporated in England and Wales. The Symbian foundation was registered at Companies House in London, as an entirely new and independent company on August 28, 2008. The assets of the Symbian Foundation are owned by its members. It is governed by an independent board of founding members, comprised in the Board of Directors, each of whom holds one equal vote (www.Symbian.org).

The foundation currently has 184 members across several industry sectors: games, enterprise, communication (internet voice), finance, media/content, tools, consulting technology and outsourcing services, network operator/channel, professional services, equip-
Coordination Dynamics in Open-Source Based Platforms

The first step to become a member of the Symbian Foundation is to complete an electronic Membership Application Form. Application Form consists of four short sections: Company Information, Member Contact, Details, Member Directory, and Billing Details. Application to the Symbian foundation must be accompanied by the signed Legal Agreement(s).

Membership of the Symbian Foundation is available to organizations meeting the following criteria:

1. Must be a firm, company, corporation, partnership, association or university or other similar entity (excluding individuals).
2. Individuals can not currently become members. We are working on a program that will shortly allow individuals to participate.
3. If applying companies have a parent company, that parent company should sign on behalf of all controlled entities.
4. If there are reasonable reasons why a parent company may not become a member, then the parent company must co-sign the Deed of Adherence in conjunction with the applicant company.

5. Successful submission of the Membership Application Form.

6. Receipt by Symbian Foundation of signed Deed of Adherence.

7. Members also wishing to contribute source code or other artifacts are further required to sign a Member Contribution Agreement.

8. Approval of membership by the Symbian Foundation.

9. Receipt of payment of the annual membership fee of $1500 USD + 17.5% VAT (where applicable).

4.2.1 The Core

In the middle of the organization of the Symbian foundation is the Symbian OS, the very platform that arrays an important number of firms in the mobile industry. This OS is written in C++ and is represented graphically by the System Model (see figure 7). The layers in the system model represent the fundamental structure of the system itself, and the basic layering should remain constant between releases of the platform. A significant change in the layering between releases would imply major architectural change or a significant re-interpretation of the architecture.


“It is a complex software platform, the system model is a representation of all that we have, obviously in a structure way, we have layers, within layers we have packages, packages have collections and collections have components. The classification determines the hierarchy, which helps in determining the dependencies and helps to manage the platform. From the build point of view, it allows you to know what needs to be build. Each package is build independently but you need to determine the order of those packages needs to be build. The system is a graphical representation of the platform”.
There are currently three layers within the main Symbian Foundation platform (the Device Platform): the OS layer, middleware layer and the application layer (www.symbian.developer.org):

**OS layer:** The OS layer includes the hardware adaptation layer (HAL) required to support a specific hardware platform and which abstracts all higher layers from actual hardware and the Symbian kernel including physical and logical device drivers. It also includes low-level OS services such as frameworks, libraries and utilities, which turn the abstracted hardware and OS mechanisms into a programmable interface. It also provides all higher-level OS services across a full range of technology domains such as communications, networking, graphics, multimedia and so on.

**Middleware layer:** The middleware layer represents the functionality that is independent of UI and applications. It provides services to applications (that is, programs with which the user can interact through a user interface) and other higher-level programs. It is independent of hardware and uses the hardware-abstracted services provided by the OS layer.

Services in this layer can be specific application technology such as messaging and multimedia, or more general device services such as web services, security, device management, IP services and so on.

**Application layer:** This layer contains all the applications available as part of the Symbian platform, such as the organizer application suite, multimedia applications, telephony and IP applications, and applications for controlling device settings and so on.

Many applications also provide programmatic interfaces to allow other applications programs to access their functionality, to support extensibility or customization.

**Technology Domains**

The Symbian Foundation groups its platform packages into technology domains, based on the following factors:

- Packages that have technical relationships to each other and whose evolution may therefore be linked
• Domains that are relevant to the various stakeholders in the Symbian Foundation community.

Technology domains provide an overview of the range of packages relevant to the particular functional area of the platform you are working on or the types of software you are developing.

Each technology domain has its own roadmap, which provides information at a granularity that is in between the high-level platform plan platform roadmaps and the very detailed package roadmaps. The foundation has a team of technology managers who create the technology domain roadmaps by aggregating information from the package roadmaps.

**Packages:** Packages are the first-level decomposition of layers. Packages are logically grouped modular collections of components. Packages are owned and maintained by a package owner, a named individual from an organization member of the Symbian Foundation, but with contributions from the wider community of the Symbian Foundation. Packages are the overall responsibility of the package owner, and further details are published on the Symbian developer website.

**Collection:** Collection is a set of collaborating components that together deliver a complete, discrete, and identifiable part of the system functionality.

The collections of the baseline system model are derived from an analysis of existing platform functionality to attempt to define plausible architectural groupings. More rigorous criteria may be applied in future, as the system evolves and new functionality is added. Collections have been defined in terms of coupling between components and source tree organization (that is, common first-level directories).

**Components:** A component is the smallest architectural unit of the system. A component is understood as an implementation unit that provides a discrete, reusable piece of the system.

In concrete terms, a component is identified as a single package of binaries, data, documentation, tests and source code. A component will be defined by a single component definition file (bld.inf), which may include one or more project-definition files (.mmp).
This allows organization of the source files to correspond to the logical notions introduced by the model and hence use the model for various practical use-cases (such as management of source).

Figure 7: The System Model (www.symbian.org)
4.2.2 Symbian Groups

In addition to the working groups the Symbian Foundation has promoted the creation of a set of groups. The “Symbian Bug Squad” is formed by members who want to make a difference to the Symbian Platform by identifying and implementing improvements proposed by the community (www.symbian.developer.org). The main focus of the Bug Squad is to find, locate and fix bugs in key areas of the Symbian Platform by:

- Organising testing and test days and bug fixing weeks around critical platform areas to yield the best possible results.
- Matching contributors with varying skills to suitable tasks and in effective teams, for instance by matching bugs with programmers and pairing programmers who fix bugs with testers that can validate the fix.
- Facilitating the acceptance of contributions into the platform by making sure the results of the Bug Squad are valuable and by building relations with the Package Owners that integrate fixes.
- Rewarding Bug Squad members for their effort through the Gold and Gecko system, which can translate into presents and career help.

“Wild Ducks” Project was set out to build their own phone, with open off-the-shelf hardware (Beagle Board + modem), and the latest open source Symbian (www.symbian.developer.org). The project is integrated by Symbian specialists, members and developers. The project has a number of goals:

- Provide a complete hardware and software stack for the community to build on and experiment with,
- Keep the hardware cost as low as possible,
- Document the entire process, so the work is reproducible by community members,
- Provide a foundation for further community projects and build community relations.

The project is administered by a project manager, the group has a to-do list and scheduled activities in order to accomplish their goals. The project at the moment of the study had a telephony stack working on the Symbian^3 emulator able to make and receive calls and texts.
Another groups is the “Freedom Fighters”, which is a software incubation project that seeks to allow people to compile the Symbian platform with a free software compiler, namely GCC. By doing so, we give people the freedom to develop the Symbian platform without the need to use a commercial compiler: thus the freedom fighter connection (www.symbian.developer.org).

In more detail, the goals of the project are:

- Provide a fully open source compiler run-time (including support for floating point)
- Ensure that the kernel can be compiled with GCC 4.4
- Remove syntax errors from the Symbian code base in such a way that other Symbian builds are not broken
- Validate that the GCC build for Symbian works
- Work with the GCC community to ensure that Symbian works out-of-the-box with the latest GCC release

4.2.3 Operations

The foundation operates as a meritocracy based on contribution, holding annual elections to appoint members to the Board of Directors. The foundation operating expenses are shared equally among the device manufacturers on the board.

Four community-driven councils have been set up to guide future directions and development (www.symbian.org). These four councils are:

1. **Features and Roadmap council** - sets the overall plan and direction of the platform and tools roadmaps, based on input and contributions from the community
2. **Architecture council** - guarantees the overall integrity of the platform by working closely with contributors to the platform and reviewing technical proposals. This council is tasked with maintaining compatibility across platform releases
3. **User Interface council** - works to ensure that devices based on the Symbian platform deliver the best user experience by developing guidelines and making recommendations
4. **Release council** - coordinates the integration of community contributions into stable and timely platform and tools releases

Together the councils constitute the decision-making bodies governing the contents and specifications of foundation software releases. The councils also act as caretakers and improvers of the foundation community’s collaboration process.

The councils can, in turn, set up temporary working groups to investigate and advise on specific issues. The members of the Board of Directors have seats in each council, with additional seats available for other foundation members. The foundation will operate as a meritocracy, with board and council membership allotted based on contribution to the platform.

The Symbian Foundation Councils exist to support the foundation community and grow the competitiveness of the Symbian platform by:

- Identifying high-level market, user and technical requirements;
- Soliciting contributions that address those requirements;
- Coordinating community contributions into regular platform releases; and
- Providing transparency for all community members regarding future platform developments.

The Symbian Foundation Council Charters document describes the collective and individual functioning of the councils. It covers common issues such as membership and chairing plus the key objectives and decision-making responsibilities of each Council. Each council has its own distinct set of responsibilities, which together are designed to complement each other and provide coordinated support for contributors and community members.

In addition to the councils, the foundation is administrated by the “Leadership Team” which is divided at the strategy level in Global Alliances, Community Engagement and Technology Delivery. Under the Executive Director are the areas of Operations, Human Resources, Legal and Acquisitions (See figure 8). Lee Williams chairs the Symbian Foundation board since 2008.

Lee Williams, Symbian Foundation Executive Director, on the role of the Leadership Team:
Coordination Dynamics in Open-Source Based Platforms

"...enable the foundation to continue its efforts towards growing a thriving ecosystem, comprising a strong network of developers, partners, manufacturers and operators, around the Symbian platform."

**Figure 8:** Leadership Team Chart (adapted from: www.symbian.org)

### 4.2.4 Licensing Symbian

The Symbian Foundation has decided to use the Eclipse Public License (EPL) which is a well known and proven open-source license that includes a copyleft clause. The spirit of the EPL allows individuals or organizations to distribute the software or products built on the software and to modify it.

The software may be used or distributed free of charge but there is no warranty provided by the distributors. Most copyleft clauses apply to 'derivative works'; this is a legal term to be interpreted under the same copyright law that applies to literary works. In general, it is intended that someone can write new closed-source applications that link to Symbian platform libraries and closed-source plug-ins for Symbian platform frameworks. If someone modifies Symbian platform source code directly, creating a derivative work, and distribute binaries then they must release the source code under the EPL as well.

### 4.3 The Platform Release Process

Daniel Rubio, Chief Architect, Symbian Foundation:

“To put something in perspective in terms of scale we are talking about an asset of about 20 million nodes of code, that is something a single developer
can’t do on its own. It is a large contribution from many companies, in different scales. We have the large contributors which give you a 5 millions of code a quarter of the platform to the individual at home that is only going to make a few thousands line of contributions”.

The platform release process is the process by which contributors propose features for the next generation of the Symbian OS, are analyzed and if accepted by the councils will be part of the “Platform Roadmap” for the next generation release. Figure 9 presents an overview of the flow between the Councils and the broader community:

**Figure 9:** Platform release process (adapted from: www.symbian.developer.org)

The process, with each step corresponding to a number in Figure 8, is as follows:

1. A potential contributor who wants to propose a new feature for a platform release prepares a Development Proposal. This describes the market needs and benefits of the proposed development, the proposed timing of deliverables, key architectural features and any dependencies on third-party intellectual property.
2. The Feature and Roadmap Council reviews the Development Proposal, provides feedback and holds a vote to gauge support. If a majority of the council supports the proposal, it is classed as Approved and added to the scope of the relevant platform release(s). Otherwise it is classed as Rejected. The development described in a Re-
jected proposal may still be contributed to the foundation but shall not be included in the scope of an official platform release.

3. The Feature and Roadmap Council publishes a Platform Roadmap describing the contents of future releases.

4. Following the approval of a Development Proposal, the contributor prepares a Technical Solution Description, describing the high-level design for the development, including architectural impact, security considerations and any proposed additions or changes to APIs.

5. The Architecture Council reviews the Technical Solution Description, provides feedback and holds a vote to gauge support. If a majority of the council supports the proposal, it is classed as Approved. Otherwise, it is classed as Rejected.

6. If the proposed development has a significant impact on the user experience, the contributor is asked to prepare a UI concept proposal describing the high-level user interface design.

7. The User Interface Council reviews the UI Concept Proposal, provides feedback and holds a vote to gauge support. If a majority of the council supports the proposal it shall be classed as Approved. Otherwise the proposal shall be classed as Rejected. Developments corresponding with Rejected UI Concept Proposals shall be removed from the scope of the relevant platform release(s). Upon rejection of a UI Concept Proposal, the User Interface Council Chair shall notify the other council chairs.

8. Where a Development Proposal has been approved for inclusion into a given platform release, the relevant Release Project Manager shall work with the contributor to obtain information regarding the scheduling of the proposed contributions. The Release Project Manager shall use this information to prepare (or update) the Release Project Plan, highlighting the status and risks associated with each deliverable in the release.

9. The Release Council reviews the Release Project Plan, provides feedback and votes to approve or Reject the plan. Where a majority supports the plan, it is classed as approved, otherwise it is classed as Rejected. Accordingly, the council may issue recommendations to the Feature and Roadmap Council or Architecture Council to make modifications to the Roadmap or Architecture if necessary.
4.3.1 Symbian Release Planning

The Symbian Release Planning is a compound of plans that the Foundation together with the different members and actors develop together to provide a common understanding and basis for agreeing: what will be delivered, when it gets delivered and how it gets delivered and then to manage the execution of releases against these plans to a successful outcome.

This starts from the Platform Plan (or Roadmap) owned by the Platform Propositions team and reviewed with the Features & Roadmap Council, and develops into the set of plans owned by the Release Management team and reviewed with the Release Council. These include the Platform Release Plan, the Integration Plan, the Test and Quality Plan and the Kits Schedule.

Developers engaged in contributing to the development of the next version of the Symbian OS have the “Road Map” or “Release Plan” as the master plan. The road map is a document provided by the foundation that comprehends the evolution of the experiences of users, developers and device creators by indicating the new features, technologies and characteristics the Symbian foundation will include in its next version of the OS. The document also includes projected release times of the coming versions of the OS (see figure 10).

The Integration Plan contains selected package features from the package backlogs which are followed closely by the Release Council in order to understand the status of the releases and the availability of content in the kits. The aim is to collect information provided by the package owners and use it to analyze the health of the release.
A Key Feature (KF) is a high level feature that has been selected by the Release Council as critical for a particular release, for one or more of the reasons below:

- The feature has a large or complex integration
- The feature has a pervasive impact on the platform
- The feature is a headline selling point for the platform that raises interest in the community. This is highlighted by the F&R Council
- The features is requested to be added by the release council

An important element of the integration plan is the “integration plan burndown chart” (see figure xx) which represents the current contribution intent for key features to be integrated into the Symbian SCM system.
Figure 11: Integration plan burndown chart (adapted from: www.symbian.developer.org).

The integration plan is based on voluntary contributions, and the aim is to increase confidence in it by promoting frequent stage deliveries and asking contributors to provide regular updates when changes occur. The integration plan has a regular “two-week heartbeat” for the kit releases and tracking feature increments. (See figure xx)

Figure 12: Sample of the Integration Plan’s “Two-week heartbeat” (adapted from: www.symbian.developer.org)

The quality plan document describes some Quality-related metrics that Symbian Foundation is considering for implementation. The document includes historical “snapshots” of the OS state. Quality of the OS is monitored by a working group (WG) and it order to assure quality it perform tests to the Symbian OS. A working group receives its mandate from that council (or exceptionally from the Board of Directors) where that council wishes to delegate work to a more focused group with the time and skills to dedicate to the task.
4.4 Interaction Channels

The Symbian foundation has created a number of channels to interact with its members, developers and general public who is interested on the Symbian OS and the foundation. The mechanisms are:

4.4.1 Online Ideas Exchange

Symbian Foundation launched the industry’s first ideas exchange online at www.ideas.symbian.org. The website will enable Symbian’s community and consumers worldwide to influence the future of mobile. The ideas exchange allows site visitors to submit their ideas for enhanced mobile features, apps and improvements to the Symbian Foundation itself for community validation through a voting system. Ideas well received by the community will be assigned to an expert who will work to bring them to fruition. More than 115 ideas were submitted to the Symbian Ideas site since it launched on Day 1 and more than 75 visitors to SEE 2009 submitted their ideas there and then at the Ideas stand (www.ideas.symbian.org).

Lee Williams, Executive Director of the Symbian Foundation (www.symbian.org):

“We launched the industry’s first ideas exchange, and established a new precedent for how conferences can run in the mobile marketplace,”

Ideas typically go through several stages: an idea is added to the Symbian ideas website, the community validates the idea, popular ideas are reviewed by an expert and the idea is implemented.

These stages are aligned with the lifecycle of an incubation project (www.ideas.symbian.org):

1. During the expert review phase of the idea, the expert determines if incubation is a good way to implement an idea and will advertise the idea in order to find a project lead.
2. When a project lead is found, the project lead puts a project proposal together and kicks-off an incubation project.
3. When the incubation project has achieved its goals, the project is completed and the idea is marked as done.

![Diagram: Innovation Life Cycle of Ideas at Symbian Foundation]

**Figure 13**: The innovation life cycle of Ideas at Symbian foundation.

The following pre-requisites need to be fulfilled before an incubation project can be started:

- The project needs a **Project Champion** - someone who champions the creation of the incubation project and ensures that it project fulfils the chief aims of incubation. A champion can be a Symbian Technology Manager, nominated expert, Community Manager or Council Member.

- The project needs a **Project Lead** to drive the creation of the project and lead it once it has been started. A project lead is an individual, who can - but does not need to - work for a company.

- Although not a prerequisite, we strongly recommend that a **Project Mentor** is assigned to the project. A mentor is an experienced community member or a Symbian employee who helps to guide and support the project.
4.4.2 Symbian Horizon

In October 2009 the Symbian Foundation announced the launch of Symbian Horizon, the publishing program for Symbian developers, at the Symbian Exchange and Exposition 2009 (SEE 2009). Developers can now sign up for the program to have their applications listed in the Symbian Horizon Directory, processed through Symbian Signed, and published to a growing list of application store partners. The goal of Symbian Horizon is to help developers bring their applications to the largest mobile market in the world in a cost effective way. This publishing program is now being scaled up to process thousands of applications in 2010 (www.horizon.symbian.org).

The Symbian Foundation created Horizon in response to developer concerns that there were too many challenges and costs associated with developing and publishing a Symbian application to the global market of potential customers. In addition to reducing these costs, Symbian and its members seek to improve developers’ access to global markets through translation and localization assistance.

Since the initial launch plans were announced in July 2009, Symbian has processed an initial group of 50 applications and is helping these developers sign and submit their products to mobile application stores worldwide. A total of five stores now support Symbian Horizon. Along with the initial stores announced, Ovi Store by Nokia, Samsung Applications Store and AT&T’s MEdia Mall, two new stores are now participating: China Mobile’s Mobile Market, and Sony Ericsson’s PlayNow arena.

Lee Williams, Executive Director of the Symbian Foundation, comments the launch of Symbian Horizon (www.symbian.org):

“We recognize that developers face many challenges in bringing their products to market on Symbian devices. In particular, the diversity of application stores in our ecosystem increases the burden on developers by requiring multiple submission and review processes. But this diversity can also offer an advantage over competitors’ closed systems, where applications sometimes receive arbitrary or commercially motivated rejections. Symbian Horizon retains this advantage while reducing the burden by becoming a conduit to
multiple stores, helping developers reach the largest global mobile market in the world more efficiently."

Yong-suk Moon, Vice President and Managing Director of Samsung Electronics Research Institute based in UK, comments on the launch of Symbian Horizon (www.symbian.org):

"I expect that Symbian developers to have broader opportunities to reach consumers at the Samsung Applications Store thanks to Symbian Horizon. We welcomes a valuable effort of Symbian Foundation and will support Symbian Horizon to provide benefits of total eco-system partnership in the industry."

There are three ways developers will be able to participate in the publishing program. Available today, all developers can have their Symbian Signed applications appear in the Symbian Horizon Directory free of charge. The goal of the directory is to provide a complete guide to every Symbian Signed application, as well as where they can be downloaded or purchased. The directory is live in beta, showcasing an initial group of applications and allowing developers to submit other applications that have already been signed.

Secondly, developers will be able to register online and edit their application and developer profiles, providing both stores and consumers with complete information about their products.

Finally, the Symbian Horizon publisher program will begin to increase the number of applications processed through Symbian Signed and submitted to application store partners in the future.

Symbian is hosting conversation (on ideas.symbian.org) to explore options for funding the program. This is a critical issue for the Symbian ecosystem to address and the solution will ultimately depend upon collaboration from many contributors, including existing Symbian members, application stores, and the developers themselves. The plan for this program will be developed collaboratively with the input from all of these constituents.
4.4.3 The Symbian Blog

The Symbian blog (blog.symbian.org) is the blog specialized in releasing news about new developments and events of new applications, coming events, design, dialogue, leisure, life-hack, mobile business and tech themes. It is also a channel to communicate the values of the foundation, reminders and link to different pages related with the Symbian OS.

4.4.4 Symbian Developer

On developer.symbian.org webpage there is information for members of the Symbian app developer community (www.symbian.developer.org). Besides full access to the code itself (to download or to search and browse) developers can find:

- Product and Application Development Kits
- The tools to create, edit, build, debug, test and install software
- Code examples
- Community behavior guidelines
- Books

Anyone can become a member of the developer’s community and make contributions to the Symbian Platform. According to its type, the code contributions are divided into four levels; all of which have an associated process that must be followed before a contribution can be made:

- **Fix**: when developers want to contribute source code to fix a platform defect
- **Enhance**: when developers want to contribute source code that makes a minor enhancement to existing functionality
- **Extend**: when developers want to contribute source code that represents a major change to existing functionality or an extension to the platform
- **Invent**: when developers want to start a community-based project to create something new and innovative on the Symbian Foundation platform.

Developers are required to contribute code using the Symbian Foundation Source Configuration Management (SCM) tool Mercurial (Hg. The process for getting set up with Mercurial (Hg) is detailed in our handy Mercurial Quick Start guide.
During development, developers will be required to provide one or more of the following documents to the councils.

- Major Contribution Proposal
- Technical Solution Description
- UI Concept Proposal

### 4.4.5 Symbien Signed

Symbian Signed is the signing program administered by the Symbian Foundation on behalf of the community. If any developer is writing an application which is going to be deployed on Symbian-based phones, then it has to be signed by Symbian Signed. Developers can access to this program through the website symbiansigned.com. "Symbian Signed" is the process of encoding a tamper-proof digital certificate into an application. The certificate identifies the origin of the application by including information on the Publisher ID used during the signing process. Because the application origin is known, once an application is signed, it can use more sensitive features of the platform (www.symbiansigned.com).

"Symbian Signed" is the process of encoding a tamper-proof digital certificate into an application. The certificate identifies the origin of the application by including information on the Publisher ID used during the signing process. Because the application origin is known, once an application is signed, it can use more sensitive features of the platform. An application whose origin is unknown - ie. one which has not been signed - will not be able to access this sensitive functionality and may not even install on the device depending on the security settings installed by the manufacturer. Sensitive APIs with the Symbian Platform are protected with capabilities and the capabilities required by your application will determine which signing option is right for you. If you are writing a straight-forward application, then you should be able to avoid using APIs protected by capabilities. This will simplify what you need to do to install an application, since it will only need to be self signed to be installed. You may wish to use Symbian Signed for self-signed applications - or for those designed for version of Symbian OS pre-v9, as putting the application through the Symbian Signed process will remove the warnings given during installation (www.symbiansigned.com).
5 Analysis

5.1 Structural Composition of the Symbian Platform

In the theoretical framework was discussed that when interorganizational relationships are structurally integrated, coordination is high (Bolland and Wilson, 1994) and this structured integration is achieved through the formation of decentralized institutions (Wren, 1967; Wheten, 1981; Rosenkopft and Tushman, 1996; O’Mahoney and Ferraro, 2007).

The arguments presented in the theoretical framework and in proposition 1 are consistent with the findings in the Symbian Platform where after a period of not having communication among the firms and increasing industry fragmentation. On June 2004, Nokia, Sony Ericsson, Motorola and NTT DOCOMO announced that their intended to unite Symbian OS, S60, UIQ and MOAP(S) to create one open mobile software platform. Together with AT&T, LG Electronics, Samsung Electronics, STMicroelectronics, Texas Instruments and Vodafone they established the Symbian Foundation with formal offices in London, Helsinki and Tokyo. The foundation was structured in four councils: the feature and road map council, architecture council, user interface council and release council. The members of the Symbian Foundation elect their head of councils through a democratic process.

The Symbian foundation was designed primarily to accelerate innovation of the Symbian OS, enabling greater services and applications by promoting collaboration, technology contributions and active participation of both firm members and developers. This was conceived to reduce the problem of industry fragmentation and keeping the platform consistent.

In contrast to Ring and Van de Ven’s (1994) arguments that formal coordination evolved from informal coordination, in the Symbian Platform case it can be appreciated that the first step for the members towards coordination was to establish the formal structure of coordination, the Symbian Foundation. The founding members delegated coordination responsibilities to the new structure, giving the authority to the Symbian Foundation to develop the meanings to achieve coordination. The founding members had no more influence on the platform release process, on the contrary they aligned their efforts according to the processes dictated by the Foundation and coordinated through the coordination mechanisms developed by the foundation. Therefore in contrast to platform leadership where one firm
leads the evolution of the platform, in the Symbian Platform is through democratic processes that the evolution of the platform is decided.

Members of the Symbian Platform managed the complex interorganizational relationships by creating the Symbian Foundation and once in place the Symbian Foundation promoted the complexity of the platform by negotiating the inclusion of more members to the platform. Therefore the institutionalization of the coordination in open-source based platforms not just respond to the need to manage complexity to take advantage of a bigger synchronized pool of resources but allows the platform to increase its complexity in number of members and coordinate them through the institutionalization of coordination.

The shift of coordination activities and the pursue of common goals to a decentralized institution in the case of the Symbian foundation fits within the definitions of institutionalized coordination (Wren, 1967; Ring and Van de Ven, 1994). Moreover the emphasis on accelerating innovation by the foundation is consistent with Rosenkopft and Tushman (1996) arguments that institutions are created in industries with complex technologies to adjudicate innovation and establish standards.

In addition to the initial structure as coordination entity, it was found the Symbian Foundation promoted virtual forms of organization by creating “Working Groups”. A working group receives its mandate from that council (or exceptionally from the Board of Directors) whenever a council wishes to delegate work to a more focused group with the time and skills needed for the task. For example there is a working group assigned to the analysis and adoption of SHAI (Symbian Hardware Abstraction Interface) which defines a new, consistent hardware interface at the bottom of the Symbian platform software stack that is in many areas closer to the hardware, leading to growth of the core platform and a thinner (and easier to create/maintain) adaptation layer. These finding supports proposition 3 on which it was stated firms will create virtual groups to carry interorganizational tasks related to the attainment of common goals of the interorganizational relationship..

The creation of these communities, the member’s interaction to create a common value and distributed, predominantly information-based processes fits within the characteristics of virtuality (Sieber and Griese, 1997). Moreover these communities match Byrne (1993) defi-
nition of virtual organization given that are temporary networks with no central offices, linked by information technology to share skills and access to markets.

In addition to the Symbian Foundation, the developers’ community and the “Working Groups” it was found additional structural forms which can be term as “Autonomous Groups”, because they pursue the ultimate goal of the Symbian Foundation but not the method developed by the Foundation. The autonomous groups do not participate in the process of new platform generation, however they are linked to the Symbian Foundation vision of accelerating innovation. The “Bug Squad” is one of the autonomous groups and is a group of free-lance developers, foundation, and members’ specialists, who gather together through internet to find, locate and fix bugs in key areas of the Symbian Platform. The different actors are linked through to-do lists, project management processes, milestone calendars and personal meetings.

The “Freedom Fighthers” group goal is to allow people to compile the Symbian platform with a free software compiler, namely “GCC”. By doing so, they look to give people the freedom to develop the Symbian platform without the need to use a commercial compiler: thus the freedom fighter connection. Freedom fighters project is integrated by Symbian specialists and anyone who is interested in supporting the project.

Another autonomous group is “Wild Ducks”, which was set to create a community to match Symbian open software with open hardware and create a completely open Symbian device. The project was promoted by the Symbian Foundation and is composed by Symbian Foundation, members and companies’ specialists coordinated through internet, to-do lists and project management processes.

A key factor in the autonomous groups is they are encouraged from their formation by the Symbian Foundation, this includes the participation of Symbian Foundation’s specialists in the autonomous groups, link with other members for support and involvement and promotion through the foundation’s websites. Another important element of these autonomous groups is that they look for support with the rest of the Symbian members and the developer’s community. Support in terms of participation, involvement and technology support. Therefore the autonomous groups’ performance is closely related to the support of the rest of the platform actors.
The creation of these autonomous groups has structural and leadership implications, at the beginning of the Symbian Platform as open platform the coordination responsibilities and leadership were deposited in the Symbian Foundation, the Symbian Foundation created the processes and the mechanisms to integrate and coordinate all the efforts of the different members, however this new form of structures is now also now taking parallel coordination responsibilities. Leadership in autonomous groups has been encouraged by the Symbian Foundation, moreover the Head of the Members in the Symbian Foundation mentioned the Symbian Foundation is surprised how quick the community of members is self-coordinating.

There seems to be a shift in structural composition from an institution to autonomous groups as time pass by and the Symbian Platform matures. These groups don’t seem to be temporary since they are continuously searching for new goals and receiving recommendations from the community to carry out new projects. For example the “Wild Ducks” group in addition to open hardware has already set the goal to develop video player and QT graphics acceleration.

5.2 Coordination Processes

The platform release process is the process by which the Symbian Foundation links all the different actors, from developers, members, councils, specialists and developers to create the new version of the platform. It is in this process that the vision and main goal of the firms to create the Symbian Foundation happens, the synchronized collaboration of pooled resources to create and innovate the platform.

The planning stage of the process takes place when the Feature and Road Map Council votes to accept a technology donation or a proposal for the next generation of the Symbian OS, consistent with Malone and Crowstone (1990) arguments it is in this stage that goals are identified. Once accepted, proposals and technologies become part of the Symbian OS Road Map and are published in the “Release Plan” or “Road Map” for all the actors to envision the next platform generation and the features needed to be developed, Martinez and Jarillo (1989) termed this step as strategic planning in interorganizational relationships. The interaction and participation of the members of the platform in the planning process happens in two ways one by participating in the council or by participating in proposing new features to
platform. The participation of members in the road map development helped members to share the perception or common objects (Manole and Crowstone, 1990).

The delegation process (Martinez and Jarillo, 1998) in the development of the Symbian Platforms takes place when the proposal are submitted given that the person proposing it is submitting the document “Major Contribution Proposal” which includes the detailed plan for carrying the development.

The execution stage is carried by members of the developers’ community and can be carried by three different types of contributions: fixing a platform defect, enhancements to existing functionality and through inventing. In the case of fixing and enhancement execution takes place only in the package concerned with the changes, however in the case of an innovation that involves more than one package owner then the execution will take place in several packages.

Das and Teng (1998) argued control is used to fashion activities in order to attain goals, in the platform release process the control stage is administered by the different package owners of the Symbian Platform, control is carried to assure the package is be fixed, enhanced or extended according to the plans. In addition to the package owner the contribution is supervised by members of the four councils.

The reconfiguration stage can occur at any given point of the platform release process and it is carried by the councils’ members as a result of their evaluation of the progress made and new proposals or technologies. The use of a reconfiguration stage is consistent with Zajac and Olsen (1993) who mentioned this stage usually requires changes in the strategy, such as is the case when the councils decide an approach is not consistent with the expected goals.

The platform release process should be considered a coordination process given that manages dependencies across the different members of the Foundation (Malone and Crowston (1994). The use of this process to coordinate activities in the creation of the platform is consistent with proposition 2 that suggest firms arrayed in an open-source based platform will engage which each other in a system of coordination processes that will include planning, delegation, execution, control and reconfiguration stages.
It is important to mention that with the raise of autonomous groups, new coordination processes were conceived to attain the groups’ goals. Therefore it was found a direct link between the emergence of autonomous groups to the existence of additional coordination processes. In the case of “Wild Ducks” the participants carry interorganizational research and development processes directed to develop an open hardware that matches the Symbian open source. In the case of the “Bug Squad” a monitoring and fixing process was setup to involve the members in improving the current Symbian platform. In the case of “Freedom Fighters”, the groups set up processes to develop alternative tools for developers.

5.3 Coordination Mechanisms

It was found that as formal “ex-ante” coordination mechanisms (Ittner et al., 1999; Ouchi, 1979), the Symbian Platform uses a variety of legal documents such as the member contribution agreement, membership rules, behavior guidelines, licensees and deed of adherence. These documents have as goal the adherence of members to expected behaviors that reduces uncertainty, roles definition and it makes interactions among members predictable (Ring and Van de Ven, 1994). For example in the case of package owners, it is expected to be active feature sponsors in their package area, do all they can in order to deliver their packages according to the release plan and support the release manager to create project plans - providing input on time and quality, including risks, issues and recommended actions.

Although the Symbian Foundation has develop an extensive number of formal coordination mechanisms to assure expected behaviors, reduce the knowledge curve for new members and facilitate the coordination of the development of the platform, events outside of the scope of those marked in procedures, policies and contracts need to be handle. According to Das and Teng (1998) “expost” mechanisms are exercised in the form of decision making meetings among the members, in the same sense the Symbian Foundation have council meetings to decide the actions needed to be executed in the case formal coordination mechanisms fail to shed light over the issue raised.

The main formal coordination for the Symbian Platform is the platform roadmap, because in it all of the proposals of the members are reflected in one single document that serves as
common goal. All the members of the platform can see the direction of the evolution of the Symbian platform and relate their tasks to the achievement of this common goal.

As contrary as it was considered in the theoretical framework, ICT (information and communications technology) was just not another coordination mechanism in a pool of informal mechanisms but it was found ICT plays a central role (Dess et al., 1995). The different Web-Pages of the Foundation contains all of the coordination mechanisms necessary for the coordination of the different actors at all levels. ICT links different actors in different locations to participate in the platform release process, links specialists in virtual groups, links together the autonomous groups as well.

ICT plays a strong supportive role for socialization in the Symbian open-source based platform. Members, specialists and developers actively participate in web based forums to discuss about product development on Symbian, technology domain and package, developing applications, discussions within and between councils and the community. It is here in these forums that members of the councils discuss and emit votes over the platform release process and on different issues. Moreover, these forums are used for announcing, running and discussing community projects, not to mention that there are forums in other languages such as Chinese and Japanese.

The socialization of the Symbian Platform members with each other through different blogs and forum promoted self-coordination (Wren, 1967), foundation helped to develop values and norms (Das and Teng, 1998) and the interaction lead community members to achieve trust among each other (Macneil, 1983).

It was found that ICT works as information storage for all the actors. Toolkits for developing applications and the code for contributing to the Symbian open source can be downloaded directly from the foundation’s website. Symbian developers’ through the foundation’s webpage have access to the developer’s library which contains information ranging from books, articles, diagrams, processes and tutorials. What is more, the library is open for developers’ contribution with information that might be useful to members of the foundation.

During the study it was observed that ICT served as control mechanism by allowing developers, members and councils to follow the progress of the new generation of Symbian OS
and the progress of projects within the developers’ community through the use of online charts, schedules and metrics.

Given these findings, there is sufficient evidence to support proposition 4, that is mechanisms of coordination within Symbian foundation were found to belong to both formal and informal therefore having in place a hybrid mechanism (Farrel and Saloner, 1988) to coordinate the different actors.

5.4 Intepretive Model

As it was discussed previously, there seems to be a shift in structural composition from an institution to autonomous groups as time pass by and the Symbian Platform matures. In the following section an attempt to interpret the developments in the evolution of the coordination dynamics within the Symbian Foundation will be made.

In order to analyze the developments in the Symbian Foundation the “Coordination Phases Model” is used as an interpretive tool. In it firms were classified according to the degree of dependence they have with the institutionalized coordination, coordination processes and mechanisms. The degree of control needed from the Symbian Foundation to accompany firms in the evolution of their performance and time was added as the third element.

Firms, when they first have contact with the Symbian Foundation, they rely heavily in the institution, process and mechanisms to interact with the rest of the members due to the lack of experience. In the same way the Symbian Foundation must control these new firms by assessing them and assuring the expected behavior. These firms which we will term “high dependent” are found in the lower right quadrant of the “Coordination Phases Model” and present characteristics of high dependence in the institution, the processes and coordination mechanisms to perform activities within the Foundation.

Firms in the upper right quadrant which will be call “regular dependents” are firms that with time, interaction and familiarization with the coordination dynamics of the Symbian Foundations, start reducing its dependence on the institution not so with the processes and mechanisms. The degree of control needed, control in the sense of accompanying during the process, for these firms start diminishing.
In the upper left quadrant are firms which are “low dependence”, these firms have reach a level of interaction and socialization that allow them to have a greater understanding of the platform as a whole. This can be seen in the increasing participation in the proposition of new features and technologies for the new generation of the Symbian Platform. Their propositions are accompanied by a good understanding of the implications in other packages of the platform. Firms in this quadrant therefore require less control and supervision in the execution of processes and doubts related to the implications of being part of a complex agglomeration of firms.

In the lower left quadrant of the model are located the “autonomous groups”, these groups are conformed of firms that have reached a high degree of socialization, awareness of possibilities outside of the methods developed to innovate and accelerate the Symbian platform. As a result of these firms are able to envision and carry out alternative projects to contribute to accelerate innovate within the Symbian Platform. In these cases the Symbian Foundation no longer controls the development of this projects but it encourages them.

![Figure 14: The Coordination Phases Model](image)

The “Coordination Phases Model” helps to understand the impact socialization and intensive interaction has on the development of open-source based platforms, there is to say that there is a relation between the degree of interaction and socialization with the even-
tual autonomy of the firms. According to Lauren Sarno, Head of the Members, the Symbian Foundation is pursuing the self-coordination of firms. This because it stimulates the number of projects and resources created to enhance innovation across members of the Foundation. Moreover the classification of firms can be useful for the Foundation to detect the requirements firms have to move to the next phase and drive them to create an even more dynamic platform.
6 Conclusion

The purpose of this study was to understand how unfocused and constantly evolving agglomeration of firms with different objectives, technologies, information and resources coordinate to create an open-source based platform.

In the case of the Symbian Platform, it was found that firms established the Symbian Foundation as first step towards coordination. The Symbian foundation was designed primarily to accelerate innovation of the Symbian Platform, enabling greater services and applications by promoting collaboration, technology contributions and active participation of both firm members and developers. Members created the Symbian Foundation to manage the complex interorganizational relationships and once in place the Symbian Foundation not just coordinated all the actors but promoted the complexity of the platform by negotiating the inclusion of more members to the platform.

The founding members delegated coordination responsibilities to the Foundation, giving the authority to the Symbian Foundation to develop the meanings to achieve coordination. The founding members had no more authority on the platform release process, rather the platform shifted to a democratic process settled by the Foundation.

The Symbian Foundation promoted virtual forms of organization by creating “Working Groups”. These groups are created by the councils to delegate work to a focused group with the time and skills to carry the task. In addition to the Symbian Foundation, the developers’ community and the “Working Groups” it was found “Autonomous Groups” that pursue the ultimate goal of the Symbian Foundation but not the method. These autonomous groups do not participate in the process of new platform generation neither responds to specific mandates from any of the councils, however, they are linked to the Symbian Foundation vision of accelerating innovation.

The platform release process is the process developed and established by the Symbian Foundation to link all the different actors, from developers, members, councils and Symbian Foundation’s specialists to create the new version of the Symbian Platform. It is in this process that the vision and main goal of the firms to create the Symbian Foundation happens,
the synchronized collaboration of pooled resources to create and innovate the Symbian Platform.

The Symbian Platform uses a variety of legal documents such as the member contribution agreement, membership rules, behavior guidelines, licensees and deed of adherence. These documents have as goal the adherence of members to expected behaviors that reduces uncertainty, roles definition and it makes interactions among members predictable. The main formal coordination for the Symbian Platform is the platform roadmap, because in it all of the proposals of the members are reflected in one single document that serves as common goal. All the members of the platform can see the direction of the evolution of the Symbian platform and relate their tasks to the achievement of this common goal.

In Symbian Platform ICT is a strong binding mechanism that increased the level of interaction, participation and coordination across the Symbian Platform. ICT allowed the Symbian Foundation to run its coordination processes and coordination mechanisms online and by doing so increased the level of participation, trust and accelerate the outcomes. Informal coordination mechanisms relied in the online forums and the different websites of the Foundation. Moreover, ICT worked as information storage center for all the actors, and it is used as means for controlling and monitoring the development of the platform.

In the Symbian Platform coordination was achieved through the creation of an institution, this institution in turn developed a process that links all the actors to pool knowledge and resources in order to create the platform. The Foundation heavily relied in ICT to implement a set of formal and informal coordination mechanisms to assure participation, behaviors and the vision of a common goal. In addition to this it was found the Symbian Foundation is increasingly promoting the self-coordination of the members by supporting the creation of autonomous groups.
7 Limitations

The study could have included interviews with members of the developer’s community, and autonomous groups in order to obtain a holistic perspective of the development of the coordination processes and the use of coordination mechanisms.
8 Implications

8.1 Implications for Practice

The understanding of coordination dynamics within open-source based platforms has two main implications for practice. First, it helps managers in open-source based platforms to develop a strategy that focuses firms in the attainment of common goals through the use of institutions, processes and coordination mechanisms. Secondly, the application of coordination dynamics in open-source based platforms lead to the access of a greater pool of resources that one single company couldn’t access outside of the platform.

The coordination of complex platforms such as the open-source based platforms requires strong negotiation capabilities from managers in order to reach consensus and a holistic perspective of strategy, shifting strategy outside of the boundaries of the organization.

8.2 Implications for Theory

As contrary to current literature that states platforms are driven by a platform leader in the Symbian Platform it was found a more democratic type leadership. The institutionalization of coordination in the form of a Foundation in which all the members have representation in all the decisions is an emergent phenomenon in the platform literature, therefore efforts in extending the current literature should be carried. The study of democratic form of leaderships is a rich area for further analysis, especially because complex agglomeration of firms in the form of platforms is taken an increasing relevance in today’s business arena.

8.3 Implications for further research

The Symbian Foundation promoted the existence of autonomous groups coordinating activities within firms in alternative projects but reaching the same goal. It is proposed for further research to analyze how these groups are created, under which terms are organized and the roles the members have. It is important for the understanding of coordination in open-source based platforms how alternative goals are set, how these goals gain support by the members of the platforms and how these interorganizational endeavors are organized parallel to those carried by the institution created to regulate coordination. Addressing these is-
Coordination Dynamics in Open-Source Based Platforms

...sues will allow researchers to increase the understanding of how autonomous groups contribute by alternative methods the achievement of the common goals.

Although autonomous groups in nature carry different processes they all converge in a common interorganizational process which is evangelization of their projects in order to obtain support from the members and developers community. Therefore for further research this process should be considered when studying coordination processes within open-source based platforms from an autonomous group’s perspective.
9 References:


Boudrean and Hagiu (year have to change in TFW) book.


Canalys 2010, canalys.com (Retrieved on 22 May, 2010).


Coordination Dynamics in Open-Source Based Platforms


Coordination Dynamics in Open-Source Based Platforms


Coordination Dynamics in Open-Source Based Platforms


Gilson, Sabel and Scott. 2009 Contracting for Innovation: Vertical Desintegration and Interfirm.


http://developer.symbian.org/

http://ideas.symbian.org/homepagelight

http://ideas.symbian.org/homepagelight

http://blog.symbian.org/

https://www.symbiansigned.com/app/page


Imai K, Baba Y., 1989. Systemic innovation and cross-border networks: transcending markets and hierarchies to create a new techno-economic system - Institute of Business Research


Coordination Dynamics in Open-Source Based Platforms


**Coordination Dynamics in Open-Source Based Platforms**


Coordination Dynamics in Open-Source Based Platforms


Coordination Dynamics in Open-Source Based Platforms

Whitney, Daniel (chair) and the ESD Architecture Committee (2004), 'The influence of architecture in engineering systems,' Engineering Systems Monograph, MIT Engineering Systems Division, March.


www.symbian.org

