Applying Hierarchical Feature Modeling in Automotive Industry

Olesia Oliinyk
This thesis is submitted to the School of Engineering at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Software Engineering. The thesis is equivalent to 20 weeks of full time studies.

Contact Information:
Author:
Olesia Oliinyk
Address: Gerhart-Hauptmann Str 24, 67663 Kaiserslautern, Germany
E-mail: olesia.oliinyk@gmail.com

External advisors:
Manfred Schölzke
ADAM OPEL AG
Address: Rüsselsheim, Germany
Phone: +49 6142 771264

Martin Becker
Fraunhofer Institute of Experimental Software Engineering
Address: Kaiserslautern, Germany
Phone:

Sören Schneickert
Fraunhofer Institute of Experimental Software Engineering
Address: Kaiserslautern, Germany
Phone:

University advisor:
Kai Petersen
School of Computing, BTH

School of Computing
Blekinge Institute of Technology
SE-371 79 Karlskrona
Sweden

Internet : www.bth.se/com
Phone : +46 455 38 50 00
Fax : +46 455 38 50 57
ABSTRACT

Context. Variability management (VM) in automotive domain is a promising solution to reduction of complexity. Feature modeling, as starting point of VM, deals with analysis and representation of available features in terms of commonalities and variabilities. The work is done in the context of an automotive industry – Adam Opel AG.

Objectives. This work studies the automotive specific problems with respect to feature modeling, investigates what decomposition and structuring approaches exist in literature, and which one of them satisfies the industrial requirements. The approach to feature modeling is synthesized, evaluated and documented.

Methods. In this work a case study with survey and literature review is performed. Survey uses semi structured interview and workshops as data collection methods. Systematic review includes articles from Compendex, Inspec, IEEE Xplore, ACM Digital Library, Science Direct and Engineering Village. Approach selection is based on mapping requirements against discovered approaches and discussion with industry practitioner on the regular meetings. Evaluation is proposed according to Goal Question Metric paradigm.

Results. The approach that can be followed in the case organization is described and evaluated. The reasoning behind feature modeling approach construction and selection can be generalized for other organizations as well.

Conclusions. We conclude that there is no perfect approach that would solve all the problems connected to automotive software. However, structuring approaches can be complementary and while combining give a good results. Tool support that integrates into the whole development cycle is important, as the amount of information cannot be processed using simple feature modeling tools. There is a need for further investigation in both directions – tool support and structuring approaches. The tactics that are proposed here should be introduced in organizations and formally evaluated.

Keywords: feature modeling, automotive, variability management, guidelines.
# CONTENTS

APPLYING HIERARCHICAL FEATURE MODELING IN AUTOMOTIVE INDUSTRY ........I

ABSTRACT .........................................................................................................................I

CONTENTS ............................................................................................................................II

LIST OF FIGURES .................................................................................................................V

LIST OF TABLES ....................................................................................................................VII

1 INTRODUCTION .............................................................................................................1

1.1 GOALS AND OBJECTIVES .........................................................................................2

1.2 EXPECTED OUTCOMES ..............................................................................................2

1.3 RESEARCH QUESTIONS ...............................................................................................3

2 BACKGROUND ...............................................................................................................4

2.1 BASICS OF AUTOMOTIVE SYSTEMS ENGINEERING .................................................4

  2.1.1 Characteristics of Products ....................................................................................4

  2.1.2 Characteristics of System Architecture ................................................................5

  2.1.3 Challenges of Automotive Domain ........................................................................6

2.2 BASICS OF SOFTWARE PRODUCT LINES ................................................................7

3 RELATED WORK ............................................................................................................9

4 RESEARCH METHODOLOGY .......................................................................................11

4.1 OVERVIEW OF THE RESEARCH METHODS ..............................................................11

4.2 RESEARCH METHODS SELECTION .........................................................................12

  4.2.1 Research Question 1 .............................................................................................13

  4.2.2 Research Question 2 .............................................................................................13

  4.2.3 Research Question 3 .............................................................................................13

  4.2.4 Research Question 4 .............................................................................................14

5 STATE OF THE PRACTICE OF FEATURE MODELING IN AUTOMOTIVE DOMAIN ..15

5.1 RESEARCH METHOD ..................................................................................................15

  5.1.1 Sample/Subjects .....................................................................................................15

  5.1.2 Data Collection .....................................................................................................17

  5.1.3 Data Analysis .......................................................................................................18

  5.1.4 Threats to Validity ................................................................................................19

5.2 INTERVIEW RESULTS ................................................................................................21

  5.2.1 Current Situation with respect to Product Line Engineering ................................21

  5.2.2 Problems with the Current Approach ....................................................................24

  5.2.3 Challenges in Adopting PLE ...............................................................................26
5.2.4 Stakeholder Expectations to the Future Approach .......................................................... 27
5.3 STAKEHOLDER GOALS ....................................................................................................... 28
  5.3.1 Improvement Goals. QFD .......................................................................................... 30
  5.3.2 Improvement Goals. GQM ....................................................................................... 34
5.4 EVOLUTION SCENARIO ................................................................................................... 40
5.5 SUMMARY ......................................................................................................................... 40
6 STATE OF THE ART ............................................................................................................. 45
  6.1 RESEARCH METHOD ....................................................................................................... 45
    6.1.1 Search ....................................................................................................................... 46
    6.1.2 Selection of Papers .................................................................................................. 47
    6.1.3 Data Collection ....................................................................................................... 47
  6.2 RESULTS .......................................................................................................................... 48
    6.2.1 Literature Reviews .................................................................................................. 48
    6.2.2 Feature Modeling Notations .................................................................................... 49
    6.2.3 Guidelines ............................................................................................................... 50
    6.2.4 Approaches for Structuring Large Scale Product Lines ........................................... 52
    6.2.5 Problems and Evolution of Feature Models ............................................................ 57
  6.3 DISCUSSION AND SUMMARY ...................................................................................... 57
7 INDUSTRY EVALUATION OF FEATURE MODELING APPROACH .................................. 59
  7.1 RESEARCH METHOD ....................................................................................................... 59
  7.2 APPROACHES TO STRUCTURING FEATURE MODELS ................................................ 60
    7.2.1 Monolithic Feature Model ...................................................................................... 60
    7.2.2 Modular Feature Models ....................................................................................... 62
    7.2.3 Multiple Product Lines .......................................................................................... 63
    7.2.4 Conclusion ............................................................................................................... 64
  7.3 TACTICS .......................................................................................................................... 65
    7.3.1 By Functional Architecture .................................................................................... 65
    7.3.2 By Organizational structure .................................................................................... 67
    7.3.3 Mirroring the Solution Space Structure .................................................................. 68
    7.3.4 By Asset Type in the Domain .................................................................................. 69
    7.3.5 Focus on the Market Needs ..................................................................................... 69
    7.3.6 By Physical Structure ............................................................................................. 70
    7.3.7 By Stakeholder Concerns or View Based Decomposition ...................................... 71
    7.3.8 Evolution and Localization of Change ..................................................................... 71
    7.3.9 Using Context Variability ...................................................................................... 71
    7.3.10 By Feature Category ............................................................................................. 72
    7.3.11 Conclusion .............................................................................................................. 72
  7.4 FEATURE MODELING APPROACH WITH GEARS ........................................................ 73
    7.4.1 GEARS Modeling Concepts .................................................................................... 73
LIST OF FIGURES
Figure 1 Thesis Structure .............................................................................................................. 5
Figure 2 Different Layers of Abstraction in Automotive Architecture [14] ..................................... 6
Figure 3 The Software Product Line Engineering Framework [49] .................................................. 8
Figure 4 Requirements Sources .................................................................................................... 15
Figure 5 Interview Process .......................................................................................................... 22
Figure 6. GM tool chain today and tomorrow .............................................................................. 22
Figure 7. V-Model ......................................................................................................................... 22
Figure 8 Problem Interdependencies ............................................................................................ 27
Figure 9 Traceability between common artifacts .......................................................................... 27
Figure 10 Traceability of variability between artifacts ................................................................... 27
Figure 11 QFD Template .............................................................................................................. 28
Figure 12 GQM Structure [63] .................................................................................................... 30
Figure 13 Search Process .............................................................................................................. 50
Figure 14 Feature Categories [43] ................................................................................................. 51
Figure 15 A feature model example: PBX product line [43] ............................................................ 51
Figure 16 Common feature model - for all vehicle lines [16] ........................................................... 51
Figure 17 Assignment dependency and example of mandatory, optional and alternative dependency [16] ................................................................................................................. 53
Figure 18 Example of multi-level tree [51] .................................................................................. 53
Figure 19 Hierarchical decomposition and subset structure [60] .................................................... 54
Figure 20 Multiple product line feature diagram of the infotainment domain [31] ......................... 54
Figure 21 Literature Review Mind Map ......................................................................................... 55
Figure 22 Monolithic Feature Model (RE – requirements engineering artifacts, D- design artifacts, QA- quality assurance artifacts) .................................................................................. 56
Figure 23 Modular Feature Models ............................................................................................... 62
Figure 24 Product Line of Product Lines ...................................................................................... 62
Figure 25 PL of PLs with modular decomposition ......................................................................... 62
Figure 26 Functional decomposition ............................................................................................ 65
Figure 27 Functional Decomposition in the case organization ...................................................... 65
Figure 28 Organizational Structure in the case organization ......................................................... 65
Figure 29 Decomposition according to market criteria ................................................................. 66
Figure 30 Physical structure of the Park Assist Control Module .................................................... 66
Figure 31 FM with Context Variability ........................................................................................ 67
Figure 32 GEARS main elements ................................................................................................ 67
Figure 33 Assets in GEARS .......................................................................................................... 67
Figure 34 Parking Assist .............................................................................................................. 77
Figure 35 Production Line "Vehicle" .................................................................78
Figure 36 Vehicle Products ..............................................................................78
Figure 37 Vehicle Option Codes .....................................................................79
Figure 38 Active Safety PL structure ...............................................................79
Figure 39 Object Detection feature model........................................................80
Figure 40 Active Safety products .....................................................................80
Figure 41 First model for Park Assist...............................................................81
Figure 42 PA feature model.............................................................................82
Figure 43 PA with feature categories ...............................................................83
Figure 44 Park Assist products .......................................................................83
Figure 45 Sensor Fusion Object choices ..........................................................83
Figure 46 Avoiding redundancy ....................................................................84
LIST OF TABLES

Table 1 Research methods characteristics [53,62] ................................................................. 12
Table 2 Summary of aims, research questions, outcomes and methods of master thesis ..... 14
Table 3 Interview Participants ................................................................................................. 16
Table 4 Workshop Participants ............................................................................................. 16
Table 5 Problems with the current approach ........................................................................ 25
Table 6 Customer Voices ........................................................................................................ 32
Table 7 Quality Goals Explained .......................................................................................... 34
Table 8 GQM definition of G1 .............................................................................................. 35
Table 9 Subgoals, Questions and Metrics for G1 (Understandability) .................................... 36
Table 10 GQM definition of goal G2 ..................................................................................... 36
Table 11 Subgoals, Questions and Metrics for G2 (Correctness) ............................................ 37
Table 12 GQM definition of goal G3 ...................................................................................... 37
Table 13 Subgoals, Questions and Metrics for G3 (Consistency) ............................................ 37
Table 14 GQM definition of goal G4 ...................................................................................... 38
Table 15 Subgoals, Questions and Metrics for G4 (Completeness) ........................................ 38
Table 16 GQM definition of goal G5 ...................................................................................... 38
Table 17 Subgoals, Questions and Metrics for G5 (Modularity) ............................................. 38
Table 18 GQM definition of goal G6 ...................................................................................... 39
Table 19 GQM definition of goal G7 ...................................................................................... 39
Table 20 Subgoals, Questions and Metrics for G6 (Redundancy) and G7 (Maintainability). 39
Table 21 Problems with the current approach and respective improvement goals ........... 41
Table 22 Challenges of feature modeling adoption in the case organization and respective improvement goals ................................................................. 41
Table 23 Explicitly Stated Requirements to FM ................................................................. 42
Table 24 Quality Goals Summarized ..................................................................................... 43
Table 25 Requirements to the VM modeling approach in the case organization ............... 44
Table 26 Keywords ............................................................................................................... 46
Table 27 Feature Declaration Types in GEARS [26] ............................................................. 73
Table 28 Technology implementation of selected approaches .............................................. 75
Table 29 Dependency types and GEARS Implementations ................................................ 76
Table 30 Challenges in implementation .............................................................................. 77
Table 31 Evaluation of approach by GQM ............................................................................ 89
Table 32 Requirements fulfillment ....................................................................................... 91
1 \textbf{INTRODUCTION}

Software intensive domains are becoming more and more complex. The scope of software systems in automotive industry increased drastically in recent times. While some features are standard, the others follow constant changes over the period of time. New versions appear, technologies develop, while the previous have still to be supported. These are all variabilities of the system. The amount of variability in automotive domain is expected to be high [14][15]. This domain has its peculiarities and challenges, which should be analyzed in this work. To reduce and help managing this complexity the approach of software product lines (SPL) is being applied. Analyzing and modeling variability of features is the first step in SPL approach. Chosen variability modeling technique and tool is the key to future management and evolution of the system.

Variation Management (VM) in industrial settings is still challenging [13]. The biggest problems are lack of methodological guidance in dealing with complexity [66], unclear taxonomy of domains, subsystems, features; unavailability of modeling guidelines (naming conventions, criteria for modeling variants, for modeling assertions) [62]. Research has provided some guidance, models and tools, but it is still challenging in the industrial setting. The research on guidance on how to choose VM techniques, approaches and tools in particular domain and situation is needed.

Hierarchical Variability and Feature Modeling can be one of the solutions in managing the complexity of variability that is considered promising [15][16]. Hierarchy provides benefits for such a complex domain by allowing separation of stakeholder concerns, decomposition of system into different levels of abstraction and modules for better visibility and understandability. But at the same time problems such as consistency and traceability between models should be taken into consideration. That is why hierarchical approaches to VM will be studied in this work.

This thesis work will research respective VM approaches, synthesize, apply and evaluate them in an industrial setting. Based on practical scenarios of an automotive company, respective approaches for variability/feature modeling will be identified and analyzed. Along a concrete subsystem of the company, the most promising approaches will be followed and evaluated. From these results, first guideline for variability modeling will be derived. The tool that should be used for variability management in Adam Opel AG is GEARS [49].

This work is structured as follows. First, the more detailed insight on background of automotive systems and SPL is given. Then the Research Methodology is described. Research questions, goals and expected outcomes are stated together with the selected methods. Each of the following chapters answers one of the questions and is also an input for the next one. Chapter “State of the practice of Feature Modeling in Automotive domain” discusses current state of practice of Feature Modeling in automotive industry. Results of this chapter summarize process, problems, goals and expectations for the new or improved approach. The future work is based on the stakeholder’s goals derived in this chapter.

Based on discovered problems, chapter “State of the Art” will try to find out if these problems are already addressed in literature and what are the possible solutions. The next chapter “Industry evaluation of feature modeling approach” puts the solutions found during the literature review into the context of automotive industry. Relevant approaches are discussed and the best ones are selected. This chapter also provides the GEARS specific implementation of one of the features.

Based on stakeholder’s goals the evaluation of the proposed approach is provided. The work is finished with conclusions and future work in feature modeling and variability management in automotive domain. Figure 1 shows the outline of this work.
1.1 Goals and Objectives

The aim of this thesis is formulated according to GQM (Goal/Question/Metric) paradigm [54]. The conceptual goal of this thesis work is:

To evaluate feature modeling process and feature models supporting that process with respect to suitability for automotive industry from the point of view of SPL engineer in the context of big automobile company.

The objectives for the master thesis are to:

O1. Characterize situation in industry
O2. Analyze state-of-the-art for hierarchical feature modeling approaches
O3. Identify hierarchical VM approaches for automotive industry
O4. Evaluate identified approaches with respect to usability, efficiency, maintainability, understandability, tool support, reliability, correctness
O5. Validate the results of evaluation against the goals
O6. Create the guideline on application of hierarchical feature modeling approach.

1.2 Expected Outcomes

The expected outcome of this thesis work is a guide for variation modeling and management in the field of automotive industry. Nevertheless, the results are expected to be applicable not only to selected industry domain, but also to other industries with similar requirements to variability management.

The intermediate result is implemented models with different approaches that allow further research and comparison.

Also, it will be possible to see if models provided by researchers are suitable for practical usage. It can be analyzed if available tools and techniques are good enough for real world or they are only research prototypes. Practitioner expectations will be investigated and the future research can be done in order to satisfy their needs.

So, the expected outcomes are:

1. Feature models use cases. List of requirements and challenges. Description of state of the practice
2. Literature review
3. Open points and evaluated VM models
4. Guide in VM processes

1.3 Research Questions

This thesis should answer the following research questions.

RQ1. What are the challenges and requirements of feature based software and system development in automotive industry? (Objective O1)

RQ2. What hierarchical feature modeling approaches exist in literature, what are their characteristics and how are they applied in industry? (Objective O2)

RQ3. What hierarchical feature modeling approaches satisfy the requirements of automotive industry? How the respective models look like? (Objective O3, O4, O5)

RQ4. How the respective management process should look like? (Objective O6)
2 BACKGROUND

This chapter gives a short background of automotive systems engineering and software product line engineering. It will discuss domain peculiarities and improvements connected to SPL engineering.

2.1 Basics of Automotive Systems Engineering

Software is becoming a vital part of automotive industry nowadays. The amount of software in vehicles increased from none (around 30 years ago) to almost 1 GB of code today [14]. Computer systems made it possible to realize the concept of so called “intelligent vehicle”, that can provide, for example, guidance assistance, parking assistance and traffic control [14]. The need for new functionality to satisfy customers is not the only reason for this. The other reasons are possibility to customize the cars according to individual customer needs, increased reliability and safety of the vehicle, control of fuel consumption and others [60] [14]. This increased significance of software resulted in the new challenges in automobile development [14].

This chapter will give a short overview of automotive systems-engineering process, adaptations over time and challenges that come up with the increased software involvement.

2.1.1 Characteristics of Products

Automotive software can be of different kinds covering everything from infotainment to safety critical real time control systems [60]. It can be structured according to application area, forming so called application clusters. The following can be types of software in today’s vehicles [60]:

- Multimedia, telematics and human-machine-interface (HMI)
- Body/comfort software
- Software for safety electronics
- Power train and chassis control software
- Infrastructure software

Such complex systems as automotive engineering demands a lot of non functional requirements, that include requirements for safety, reliability, quality and security even under extreme conditions for example heat, cold or accident [71]. These requirements are achieved with the help of software systems [71].

Although automotive software has a lot in common with other types of software (information systems for instance), it has a number of peculiarities:

1. The requirements for the following quality factors are usually more important [55]
   - Reliability
   - Functional safety
   - Real-time behavior
   - Minimized resource consumption
   - Robust design
   - No reboot possibility

2. Length of development cycle
3. Integration of software from different domains (e.g. infotainment, chassis)

Software development process for automotive industry should be helpful in managing such complex systems. Although, difficulties arise from high number of requirements and implementation of features, this is not the only problem. Identical functions can be provided in different versions that are relevant for some of the similar vehicles. Moreover, for proper functioning each feature needs particular ECUs, sensors, actuators, cameras and other hardware. Hardware components can also be provided in different models and series. The other complexity is keeping traceability between software and hardware units.

2.1.2 Characteristics of System Architecture

The first software in cars was used to control the engine. First functions were not connected to each other, could function independently and were deployed to separate electronic control units (ECUs) [14]. ECU is the building block of an automotive electrical and electronic system [79]. Pieces of software are deployed to the ECUs, which are interconnected through communication networks as well as connected to actuators and sensors needed to implement required functionality [79]. With such an infrastructure different ECUs could exchange data with each other that resulted in implementation of more and more cross cutting features [14]. As the system is getting complex with increased demand in functions different bus technologies were introduced [14].

According to [15] following are the aspects that are relevant for description of an automotive system (see Figure 2):

1. **Functionality Level-Users View.** This is the highest level of abstraction that captures all the features provided by the car to the user and the dependencies between those features. Users View models are helpful in understanding the overall functionality offered by the vehicle, keeping traceability between different features, documenting possible variations of the features. Feature or function hierarchies are used to model the above information.

2. **Design Level – Logical Architecture.** At this level logical architecture is described as the decomposition of functional hierarchy into components that interact with each other. Logical architecture provides an abstract solution that allows reuse of services on the conceptual level. It helps to separate implementation and deployment details from logical behavior.

3. **Cluster Level.** At the cluster level logical components are further decomposed and grouped into units of deployment.

4. **Software Architecture.** Software architecture combines description of software platforms and the applications represented by tasks. This software has to be deployed onto the hardware. The software architecture is derived from the logical architecture.

5. **Hardware Level-Hardware Architecture.** The hardware architecture should describe all the hardware elements of the vehicle and their connections. This can be sensors, actuators, bus systems, communication lines, ECUs. There are a number of specific requirements for hardware that should be fully satisfied.

6. **Deployment-Software/Hardware.** The last but not least, software should be deployed into hardware units. This mapping should be defined and together with hardware and software it represents a concrete realization of the logical architecture. The mapping between hardware and software depends on network topologies and technological requirements of functionality.
2.1.3 Challenges of Automotive Domain

So, integration of software in the automotive industry was not only beneficial, but also led to a lot of additional challenges.

Nowadays, in the struggle for customers automotive companies try to provide customizable products. For example, a car can typically have about 80 user electronic features that can be ordered by the customer. That means that there are $2^{80}$ possible variants of the car. Moreover, variants on the lower level of abstraction can achieve even bigger complexity (for example, variants of algorithm implementation) [60]. Obviously, this order of complexity is hard, even impossible, to manage. Restriction of the range of possible variants by introducing dependency rules between features, structuring vehicles according to market criteria, creating comprehensive models can be helpful techniques in managing complexity [47,48].

It can be seen that defining an automotive system is very complex. Yet another problem is selecting a tool chain that will allow connecting models on different abstraction levels and making them traceable.

According to Bühne [16] following are the challenges with the requirement specification artifacts:

- The requirements have to consider diversity of vehicles, countries, releases. The amount of requirements is huge, that makes it very hard to manage
- To correctly specify all aspects of functionality different requirement artifacts are needed: system and user scenarios, quality, data, and interface requirements, and design constraints
- To make requirements management more efficient the reuse for a new vehicle-line has to be supported
- The reuse of requirements artifacts for different vehicle-lines leads to an additional dimension of product lines, the so called multiple product lines.
At the level of requirements engineering, there is definitely a need for further research into product lines. Research remains to be done to cleanly relate feature graphs to artifacts of the design activities, as well as to identify means for efficient exploration of architectural variants.

General problem lies in the scalability of the current approaches to software development.

Software Product Line Engineering is being applied to manage the complexity of automotive development.

2.2 Basics of Software Product Lines

Along with producing standard products, which are completely the same, industry faced a need for customization of products due to diversity of customer needs. This problem was also stated before, and is very relevant for automotive industry. For example, the same car model can be produced with or without parking assistance or cruise control. This is called mass customization “the large-scale production of goods tailored to individual customers’ needs” [58]. Whereas software product line (SPL) engineering is “a paradigm to develop software applications (software-intensive systems and software products) using platforms and mass customization” [58]. This definition means that the company should think about common features of all the products and create a common platform. At the same time using mass customization means managing variability. Variability management is modeling of commonalities and differences of the products in a common structured way. It is also known as the process of defining, representing, evolving and tracing the variability [45]. Variability Management (VM) is one of the central questions in Software Product Line Engineering domain [4].

SPL Engineering process is built on two concepts: domain engineering and application engineering [45] (Figure 3). During domain engineering the product is analyzed for common and variable assets. The platform that consists of reusable artifacts (requirements, architecture, components, and tests) is also established during domain engineering. Very important is definition of traceability links between those artifacts [58]. During application engineering phase a specific product is being derived from assets created in domain engineering [58].

In domain engineering the variability and commonality is defined and modeled. Variability can exist in all the artifacts, not only requirements. Application engineering deals with binding this variability. The explicit choices over the modeled variability are made and specific end product is derived.
Software Product Lines is a popular and promising area nowadays. Benefits of using SPL approach is higher quality of product, shorter time-to-market, better maintainability [58]. To achieve the above stated benefits commonalities and variability should be modeled in the way comfortable to visualize, understand, maintain and implement [32,53]. These approaches to variability modeling are under study and research [67]. Variability information can be represented as a separate model (or models) or built-in into the system model [45]. Selection of the best fit approach is still a challenge [67].

Modeling of domain knowledge, description of commonalities and variability is the key step in adopting a SPL approach [53][71]. The right chosen VM approach is a success factor to the product implementation [71]. Building up a product from a set of reusable assets and steering the variable parts depends on the VM approach as well. It can vary from having a set of libraries or core modules to having a production line, where the creation of new product is done by selection of features.

Successful application of product line approach was documented for example in [71]. The authors tried to approach such requirements of automotive systems as maintainability, safety, reliability. They argue that wise application of variability management in automotive industry can raise quality of the product and reduce time to market, as there are a lot of variations among requirements between models, customers, technologies and costs [71].

Figure 3 The Software Product Line Engineering Framework [58]
3 RELATED WORK

The preliminary industrial problem can be generally formulated as a problem of managing variants of the features, components and configuration of alternative systems. This work concentrates on the problem in requirements engineering artifacts. However, the reuse approach should be incorporated in the whole lifecycle of the systems. Moreover, the goal is also to help managing the complexity of domain by bringing more structure, overview and better understanding of the system and its variants.

There is a lot of work done in Variability Management field. Different methods were proposed, as for example Orthogonal Variability Models [59], Model Driven Product Line Architectures [56], Text Based Feature Modeling(TVL) [22], Use Case Based Modeling [70], Feature Modeling and others. Most of the approaches are hard to maintain in the big automotive systems.

According to Czarnecki et al [26] variability modeling approaches are mostly either feature based or decision modeling based [30]. Feature modeling can be used for capturing both variability and commonality, whereas decision modeling captures only variability information (decision that should be taken) [26]. The minor, but important difference between FM and DM is support for hierarchy. In FM this is essential quality, but in DM it is not [26]. Hierarchy is an important notion for automotive industry.

In this work the solution was seen in Feature Modeling, as a way to structure and gather all variability information in one place, improve understandability, facilitate communication between different stakeholders and component responsible. Hierarchy plays also a great role in the solution as a mean to reduce complexity of the huge models by decomposing one model into smaller relevant pieces.

The other benefit of feature models compared to other methods is the range of their application. So in [68] the following application of FM are stated:

- modelling large domains
- managing the variability of PL products
- encapsulating system requirements
- guiding the PL development
- driving marketing decisions
- future planning
- communication between system stakeholders

Hierarchical FM was selected for investigation in this work as a backbone for the whole development lifecycle of the products.

There exists several notations for FM going back to FODA [41]. Most of them build up feature diagrams from alternative and mandatory nodes (features) and relations between them. The extensions of FODA notations are FORM [42], FeatureRSEB [35], Generative Programming [25] and others. The differences between them are concept of cardinality, attributes, and feature groups.

Feature Modeling is not a new concept in automotive product line engineering. Papers on applicability of feature oriented approaches in automotive industry are being published. The discussion on problems of complexity reduction[48], redundancy reduction and more efficient engineering is being done.

AUTOSAR framework [19] is becoming variant rich, where variants can be documented on several abstraction levels [65]. EAST-ADL is an architecture description language for
automotive systems that uses the concepts of feature modeling and variability modeling to support product families [24].

Another works include Multilevel feature trees [61], Feature Assembly [1], modeling for multi criteria product lines by Bühne et al[17,18], n-dimensional product lines [72,73]. Czarnecki claims that stepwise configuration and specialization of feature model is extremely important in industrial settings as usually choices on feature models are done by different people during different stages of development [27].

An interesting concept was proposed by Hartmann in [38] to use a special context feature model and the usual feature model with the link between them. This helps to manage variability across multiple product lines.

As an alternative to standard single feature model modular and multiple product lines were proposed for usage in complex domains [6,12,39]. These approaches will be analyzed in this work in the context of automotive industry.

Another important, but always missed question, is a structuring organization of product lines and feature models in particular. Few papers discuss this question [29,36].

More work that is related to feature modeling in automotive industry is gathered under Section 6.2.

With respect to the mentioned related work this thesis contributes as follows:

- Presents results of literature review on approaches that help to manage variability in automotive industry
- Applies these approaches in industrial setting and compares benefits that they can bring
- Combines the beneficial approaches to achieve the best out of them and evaluates them
- Structuring approaches of feature models relevant to automotive industry is gathered and analyzed.
4 RESEARCH METHODOLOGY

This chapter discussed research methods used during this work. First, the overview of the existing research methods will be provided. The second section of this chapter deals with selection of appropriate research methodology for this work.

4.1 Overview of the Research Methods

Depending on the purpose of the study the following research methods can be identified [63,75]:

1. **Experiment’s** goal is to measure influence of manipulating one variable on another variable. Experiments are usually highly controlled and done in the laboratory setting. The subjects are assigned at random to different treatments [75]. The effect of manipulation is measured and statistical analysis can be performed to understand the dependencies between variables. Usually experiments are designed to objectively compare methods, processes and techniques. As an example of controlled experiment can be comparing two different methods for inspections [75]. Important, that experiments are purely quantitative as they focus on measurements.

2. **Survey** is a collection of information from subjects before or after the introduction of new technology or method [63,75]. Surveys are usually used to study current situation and evaluate it according to qualitative criteria and samples opinion.

3. **Case Study** “is an empirical method aimed at investigating contemporary phenomena in their context” [63]. Case study investigates real project in the industrial settings [75]. Level of control is less compared to experiments. Case study may use other research methods, such as surveys or literature reviews. These methods are usually used as a data collection methods within case study [63]. The benefit of using case study over experiment to compare different methods is that by case study the scalability problem can be avoided. It is also easier to set up [75].

4. **Action Research** is closely related to the case study. Case study is an observational study, whereas action research aims also to influence or change the object of the research [63].

The purpose of study can vary as well. In [63] four types are defined:

1. Exploratory – discovering what is happening
2. Descriptive – describing a situation
3. Explanatory – trying to find out an explanation of a problem, situation, consequence
4. Improving – aims to improve studied aspect of situation or phenomenon

Table 1 shows the correspondence between research methods, objective and data types. Showed are only primary objectives and usage of particular method is not restricted to one purpose. For example, case studies can also have explanatory character.
<table>
<thead>
<tr>
<th>Methodology</th>
<th>Primary Objective</th>
<th>Primary data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Survey</td>
<td>Descriptive</td>
<td>Quantitative/Qualitative</td>
</tr>
<tr>
<td>Case study</td>
<td>Exploratory</td>
<td>Quantitative/Qualitative</td>
</tr>
<tr>
<td>Experiment</td>
<td>Explanatory</td>
<td>Quantitative</td>
</tr>
<tr>
<td>Action research</td>
<td>Improving</td>
<td>Quantitative/Qualitative</td>
</tr>
</tbody>
</table>

Table 1 Research methods characteristics [63,75]

### 4.2 Research Methods Selection

Before thinking of solution and improvements, the problem in industry should be studied in-depth. When the current situation is being characterized then it is possible to set the company’s goals. These goals can be, for example, improvement of the process or finding the solution to deal with complexity, or decreasing amount of rework and others. Based on the defined goals the respective methods should be selected. Possible options can be found in the research literature and published research. Applying selected methods and analyzing the results in order to select the best possible solution if possible. The last but not least is creating a report about the performed study.

In this section the motivation behind selection of the best fit research method will be presented. By best fit method it is possible to understand the method that can satisfy the goals of this work and help to retrieve the needed information.

The main goal of this work is to study current situation in industry with respect to product line engineering, discover problems, and propose possible solutions.

First of all, the research will take place in industrial setting to study problem in real life. With respect to discussed research methods, case study, survey and action research study phenomena in real life. That means that controlled experiment does not satisfy this first criterion. Moreover, as a large scale product line is hard to create, a controlled experiment cannot be used to study the problem.

So, the choice should be done between survey, case study and action research.

Characterization situation in industry is having descriptive character. The goal is to take a snapshot, a baseline of the situation before the proposed improvements. This means that a survey can be carried out.

However, apart from characterization the goal of the work is to find out different feature modeling approaches and to evaluate them in the industrial setting. This goal is having exploratory and improving character. Sometimes action research is referred to the case study where apart from observations some changes are being done. One of the main distinct characteristics of action research is “taking an action” and observing the results [8]. In this work actual application of the selected feature modeling approach to a whole subsystem or even whole system in the case organization and observing the results is not feasible. Switching to new technology in such a big organization like OPEL is a long process and exceeds the time frame of this work. However, during selection of feature modeling approach the smaller units of system should be modeled and discussed for suitability. Due to this limitations action research method cannot be followed. So, the choice between action research and a case study should be made in favor of case study. Case study is a suitable approach as it allows deep study of the problem in its natural context [63].

The case study will be conducted in a representative of automotive industry. Adam OPEL was selected as a suitable candidate as the company is in the transition from development without systematic reuse to software product line engineering. The company’s goal is to optimize its processes and improve structure and quality of their models and components.
OPEL is a subsidiary of General Motors Company. Because of the case organization’s size and a huge number of requirements the product line transition is problematic.

The unit of analysis of this work is a process of variability modeling and management in the case organization. This process was studied with respect to the whole system and Active Safety subsystem in particular.

The study conducted in this work can be characterized as single holistic study as it is done in the one case organization and within one context.

It was mentioned before that case study may include other research methods. That is why in this work industrial case study with different sources of gathering information will be conducted to give answers on stated research questions.

In the following subsections details according to each research question will be given.

### 4.2.1 Research Question 1

To explore the state of the practice a survey will be conducted, as discussed previously. Surveys are used as a pre-study to a more thorough investigation [75]. This exactly reflects situation in this work. Most common data collection methods available for surveys are: questionnaires and interviews [75] and workshops.

In order to choose the best fit variability management approach for the company’s context the current situation, problems and expectations should be identified. This can be done by conducting semi structured interviews. Semi structured interview were selected over structured and unstructured, as they allow free discussion, but also stick to the main prepared questions. Another part of RQ1 is identification of stakeholder goals with respect to feature modeling. For this purpose workshops were selected to use the cumulative ideas and brainstorming.

### 4.2.2 Research Question 2

To answer RQ2 systematic literature review will be conducted. SLR following Kitchenham [44] gives a specific guideline that can be followed. Nevertheless, it will not be performed as it is time consuming and is regarded as not feasible in the context of this thesis. The priority is to give feedback in time to the case organization. However, this is not possible while conducting full systematic review. During the study the search for papers will be done through SPLC conference and VM conference papers, scientific databases. The search process will be documented, references of the retrieved papers will be checked (snowball review) and summary will be written. The result of this study is the identification of a subset of models, their characteristics and level of industrial evaluation for further investigation. Through literature review a state of the art should be described.

### 4.2.3 Research Question 3

Using the information from RQ1 and RQ2 the study of variability management approaches should be performed.

In order to evaluate each of the models in the industrial context a study of different approaches will be done. A subsystem with real world requirements will be modeled with the help of each VM approach identified. During the implementation the process will be documented, together with problems that were faced during modeling.

The results of modeling should be compared and evaluated by practitioners from the point of view of usability, efficiency, maintainability, understandability, tool support, reliability, correctness of the modeling process (also based on the evolution scenarios) and correspondence to the goals stated during description of state-of-the-practice. This can be done during the regular meetings with company’s practitioner.
Based on the study results the most suitable approach will be selected.

4.2.4 Research Question 4

The expected result of the thesis is a methodology guideline on variability modeling and management. In order to reach this goal, information extracted on each of the previous steps should be documented. From this data guideline will be created that will help in variation management for the future work of automotive companies. This document will be an answer to the 4th research question.

Table 2 below shows the traceability between aims, research questions, outcomes and research methods.

<table>
<thead>
<tr>
<th>Aim</th>
<th>RQ</th>
<th>Outcome</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Characterize situation in industry</td>
<td>RQ1</td>
<td>Use cases and evolution scenarios</td>
<td>Survey</td>
</tr>
<tr>
<td>Analyze state-of-the-art for hierarchical</td>
<td>RQ2</td>
<td>State-of-the-art description</td>
<td>Systematic Literature</td>
</tr>
<tr>
<td>feature modeling approaches</td>
<td></td>
<td></td>
<td>Review</td>
</tr>
<tr>
<td>Identify hierarchical VM approaches for</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>automotive industry</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Evaluate identified approaches with</td>
<td>RQ3</td>
<td>Open points and evaluated feature models</td>
<td>Case Study of modeling</td>
</tr>
<tr>
<td>respect to usability, efficiency, maintainability, understandability, tool support, reliability, correctness</td>
<td></td>
<td></td>
<td>approaches based on the company’s subsystem</td>
</tr>
<tr>
<td>Validate the results of evaluation against</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>the goals</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create the guideline on application of</td>
<td>RQ4</td>
<td>Guideline of variability management in</td>
<td>Reporting</td>
</tr>
<tr>
<td>hierarchical feature modeling approach.</td>
<td></td>
<td>automotive industry</td>
<td></td>
</tr>
</tbody>
</table>

Table 2 Summary of aims, research questions, outcomes and methods of master thesis
5 STATE OF THE PRACTICE OF FEATURE MODELING IN AUTOMOTIVE DOMAIN

This chapter gives an overview of the current situation in the case organization with respect to feature modeling and product line engineering in general. The first goal of conducting this survey was to gather knowledge about current VM usages, model creation and change management. The second part of the survey was focused on challenges and possible improvements of the current processes. The interview results helped to understand current state of affairs and gave a picture of strong and weak points, as well as desired and planned changes within the field of VM. Understanding these items is an important prerequisite for finding and fitting approaches that can support the organization with addressing demands in variability management. To get an overview as wide as possible the interviews are therefore carried out with stakeholders of different domains and roles.

To formalize the industrial needs and goals workshops oriented to gather measurable goals were performed.

The requirements that should be satisfied with the selected FM approach are being gathered in this section. The inputs for them are results from the interviews and workshops. Figure 4 shows how the final set of requirements will be composed.

5.1 Research Method

In order to analyze current situation in industry a survey was performed. The goal was to analyze how feature modeling is used in the organization, to reveal the problems and challenges. The objective was best achieved by conducting a set of semi structured interviews, workshops, and meetings.

5.1.1 Sample/Subjects

Subjects were selected in such a way that we could get information from different stakeholders, who have different points of view on feature modeling usage and who work on different level of abstraction in system development. The data triangulation [63] was applied during selection of interviewees. We wanted to look at the problem from different points of
view. Ideally, we would need more than one person in each role to increase significance of the data. However, due to people’s availability and time constraints only five interviews were made. Experience of people with product line engineering is almost zero; they are only starting to adopt it in the organization.

Following are the stakeholders that were interviewed:

- System Configuration and Release manager – the person responsible for this role has an overview of all the processes that are done throughout the development lifecycle, as the concerned person is responsible for providing configurations and correct releases of software.
- Requirements Engineer was chosen for an interview to get insight of problem in requirements artifacts connected to SPL and feature modeling.
- Platform and Systems Engineer is responsible for program management of vehicle programs and has a high level view on the architecture and products of the whole system.
- Diagnostics and System Engineer – this role is connected to validation and diagnostics that is also an important part of the system.
- Subsystem Architect is responsible for modeling specific subsystem (feature) of the system. The concerned person works at the other level of abstraction than platform and system engineers. That is why his view on the problem was also valuable.

Table 3 shows the experience of the interviewees in their current role and with respect to SPL. Most interviewees have very limited experience with SPL. They are new to this topic and only begin to introduce it within the company. However, they are experts in their field of practice. Poor SPL knowledge was added as a threat to validity of the results.

<table>
<thead>
<tr>
<th>Id</th>
<th>Role</th>
<th>Experience in the current role</th>
<th>Experience with SPL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Configuration and Release manager</td>
<td>4 years</td>
<td>1 years</td>
</tr>
<tr>
<td>2</td>
<td>Requirements Engineer</td>
<td>4 years</td>
<td>limited</td>
</tr>
<tr>
<td>3</td>
<td>Platform and Systems Engineer</td>
<td>5 years</td>
<td>limited</td>
</tr>
<tr>
<td>4</td>
<td>Diagnostics and System Engineer</td>
<td>2 years</td>
<td>limited</td>
</tr>
<tr>
<td>5</td>
<td>Subsystem Architect</td>
<td>1,5 years</td>
<td>limited</td>
</tr>
</tbody>
</table>

Table 3 Interview Participants

The internal trainings about concepts of Software Product Lines were given before the start of the survey research. The 2 hour training in “RME (Requirements Management Engineering) & PLE Concepts Overview” and 8 hour course in “Management Requirements with IBM Rational DOORS [78]” was attended by all the interviewees. Also 2 respondents attended additional GEARS training course. Nevertheless, some of the interviewees still had troubles in recalling some of the concepts. In this case, the explanations were given during the interview by the researcher. For example, term “scoping” was defined before asking the practitioners about this process.

<table>
<thead>
<tr>
<th>Id</th>
<th>Role</th>
<th>Organisation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Configuration and Release manager</td>
<td>Opel</td>
</tr>
<tr>
<td>2</td>
<td>Department Head Embedded Systems Development</td>
<td>Fraunhofer IESE</td>
</tr>
<tr>
<td>3</td>
<td>Researcher in Department of Embedded Systems Development</td>
<td>Fraunhofer IESE</td>
</tr>
</tbody>
</table>

Table 4 Workshop Participants
The participants of the workshop were experts from Fraunhofer IESE and a practitioner from the company who is responsible to introduction of feature modeling in the organization (Table 4).

5.1.2 Data Collection

Interviews are one of the data collection methods that were used during the study. The format of the interviews was decided to be semi structured to allow free discussion, but also to make sure all the important points are covered. Interview questions were prepared beforehand. They were reviewed within the group of experts from Fraunhofer IESE (two persons) and also the practitioner from the case organization. The review with the practitioner was done to remove any ambiguities in the terminology and formulations used. For example, the notion of ‘feature’ was understood in different ways by people in industry and in research. This review also helped to understand the setting of the company more and to add additional, more detailed, questions.

Interview questions were grouped into four sections, each of which had its own topic. The sections were:

1. Introduction of interviewee and his/her roles. This section asked for the current and past experience and main responsibilities of the interviewee.

2. Current situation with respect to variability management. The questions of this section asked if variability modeling is currently done in the organization, how it is done and where documented. The information about strengths of the current approach was also discussed. This section also asked about responsible roles for variability management and evolution of the system.

3. Challenges that need to be addressed. The other sections concentrated on problems and challenges with the current method used.

4. Improvements that should be made. The last section of the interview was about stakeholder’s expectations and views on the possible solutions. This helped to get an understanding of what are the requirements to the approach and what criteria should be met in order to satisfy the stakeholder groups.

The goal of sectioning was to have more structure and to move logically from context and current processes to the future improvements.

The list of questions can be seen in Appendix 1.

After the finalization of questions they were sent to the interviewees. Making interviewees familiar with the topic of the interview was considered necessary to give participants first introduction to the context and our expectations from them.

Interviewing process was done face to face and over the phone. Conducting interviews face to face was the preferred form of communication by both of the parties, however due to the geographical location some interviews were done over the phone. All the participants allowed making an audio recording of our conversation. Of course, notes were taken as well, but full audio record is better for further analysis.

During interviews it was discovered that almost all of the participants are not familiar with variability management and feature modeling techniques. However, the conversation about current processes and challenges was extensive.

Interview process is schematically described in Figure 5.
The second part of survey, which dealt with formulation of stakeholder goals, was done in the workshop and meetings. Workshop was designed to brainstorm, discuss main issues and improvement goals and to reach the common understanding on expectations of the selected approach. For further clarifications we used regular meetings with company practitioner. Clarifications and discussions were also regularly held during weekly meetings with company practitioner. The results were gathered as short summaries (meeting notes).

As a part of triangulation the company’s documentation was studied to confirm the results of the interview [57]. This study included reviewing of some requirements specification documents, process documentation, specifications of variability, available training presentations. The purpose of this review was to understand more deeply how the development process is organized today, to understand the context of the company before interviews and to get insight on the background of interviewees in SPL. The goal was to understand what interviewees may talk about and to have basic idea on the company’s processes to make communication easier. Documentation review helped to strengthen the results of interviews.

5.1.3 Data Analysis

After the collection of data with the help of interviews it was analyzed in the following manner. The first step was data transcription and adding reviewing notes made during interviews. This text contained raw data, but structured according to questions asked. Next step was to perform open coding [23]. To help in data analysis Atlas.ti [40] software package was used. During open coding the information collected was broken down into parts and assigned a code. The assigned code defines the main idea of the analyzed piece of text.

The codes identified were:

- Stakeholder role
- Experience
- Responsibilities
- Motivation for SPL
- Problems with current approach
- Weakness of current method
- Challenges
- Strengths of current methods
- Change process
- Future usage scenario
- Ways to document variability today
- Improvements
- Benefits

So the raw text was chunked into manageable pieces of information. This allowed the next step of analysis. The data was grouped according to defined categories. Each category was reviewed and summarized to answer the formulated research questions.

The results of the goal definition workshop were put into Quality Function Deployment template (this will be discussed in detail in the following sections) which was the basis of goal formalization.

5.1.4 Threats to Validity

This section describes main threats to study validity and the strategies used to mitigate these risks. Threats will be discussed according to categories [76,77]: construct validity, internal validity, external validity and reliability (conclusion validity).

5.1.4.1 Construct Validity

According to Yin [77] construct validity deals with “establishing correct operational measures for the concepts being studied”.

During this survey the following threats to construct validity were identified:

1. *Incorrect interpretation of questions.* There is a threat that questions are not understood in the same way by interviewer and interviewee. To minimize this threat the review of the prepared questions were done with practitioner from the company, where the questions were discussed and reformulated if needed.

2. *Incorrect interpretation of answers.* All interviews were recorded for further analysis. Also the use of common terminology should minimize this threat. Moreover, the review of company’s documentation helped to make some of the answers clearer and get a more detailed understanding.

3. *Dependence on people.* The interview results are based on the answers of respondents. To mitigate the risk of getting not correct information data triangulation was used. People of five different roles were interviewed to get broader perspective on the results.

4. *Experience level of respondents.* The interviewees are not the experts in Product Line Engineering area. That is why the information regarding SPL area might be not completely correct or the concepts may be misunderstood. To mitigate the risk the concepts were discussed prior to interviews and training were provided for the respondents.

5. *Respondent’s bias.* Interviewees had a very little understanding of SPL. So their answers could be biased by the trainings they attended. As the company already chose the tool for SPL and the information provided by trainings were aimed for the tool adoption. This threat is hard to minimize, because it is a very subjective issue. Especially this kind of threat could influence the answers to the section of interview about future expectations. The interviewer tried to ask for the motivation behind the choices that the respondents made and the answers they gave.

6. *Researcher bias.* The researcher’s vision of the problem could possibly influence the flow of the interviews and data retrieved. To minimize this threat the interview results were reviewed by the practitioner in the company.
7. Selected stakeholders. As the company is still in the process of transition to the SPL methodology the product line specific roles were not established. The roles that should have been selected to interview are \[45\]: product manager, domain and application requirements engineer, domain asset manager and others. However in the case organization these roles are not defined. That is why the interviewees were selected from the available stakeholders. It is most likely that in the future the product line specific roles will be occupied by the existing people of the company. The correspondence between roles can be established. For example, domain core asset manager role can approximately correspond to release and integration manager role. Nevertheless, the main point of this study is to gather required information about current processes of the company (with respect to product line engineering and variability issues) and expectation of stakeholders for improvement. This information could be given by the selected respondents.

To sum up, the main mitigation strategies of threats to construct validity were using different sources of data collection, review of interview questions prior with company’s practitioner to gain the common understanding of questions, review of interview report after the completion of study.

5.1.4.2 Internal Validity

Internal validity is mainly important for explanatory studies, when the aim is to show causal relationship between studied concepts. This part of the work is having mainly descriptive character. However, the part of survey where the goals are defined implies relations between some goals and requirements.

To correctly interpret data and mitigate researcher bias the same interview process was followed:

1. Questions were formulated similarly for all the interview participants
2. Easy questions were asked in the beginning to make interviewee comfortable
3. Respondents were informed about their anonymity
4. Place and time of the interview was selected according to respondents preference

It is assumed that similar conditions of the interview should reduce the bias and comparison of results of interviews should be more reliable. Thus, this threat is minimized.

5.1.4.3 External Validity

External validity deals with the ability to generalize the study results to the context beyond the immediate study. The results of the survey can be generalized to the company’s boundaries, as the documentation reviewed was used companywide; selection of respondents was done to include main roles that deal with variability.

To generalize study results outside the specific company the similar study should be performed in another organization and results should be compared. However, this was not feasible.

Moreover, this threat is minimized during goals definition. Goals are defined in different levels of abstraction – quality goals and functional goals.

The results of this particular survey cannot be directly generalized and applied to some other organization. However, the companies of automotive domain are expected to face the same problems, as the main root of problems is the complexity of the domain. The interview design can be used to carry out survey in the organizations different from the studied.
5.1.4.4 Reliability

Reliability shows that data collection procedures can be repeated by other researches with the same results.

The main threats to reliability are:

1. Incorrect data interpretation by researcher. There is a risk that data is interpreted by the researcher and the other researcher might derive different conclusions from the same data. To control this risk data triangulation was used to ensure correctness of the results. The other mitigation strategy was recording the complete interviews, so that researcher can have all the required information.

2. Poorly documented study process. There is also a risk that the research procedure is poorly documented and will not allow replication of study. To mitigate this risk the survey report is reviewed by researcher from Blekinge Institute of Technology.

Overall, this threat is hard to minimize as the work was done individually. However, different data collection methods used help to minimize researches bias. Also group data collection methods as workshops and meetings help to share the ideas and clarify the gathered information.

5.2 Interview Results

The following sections will give summary of the data collected during the interviews. Each section will characterize situation in the company from the different point of view – current processes for managing variability, problems with it, challenges in adoption of PLE and improvement expectations of the practitioners. The following sections provide answers to the first research question:

RQ1. What are the challenges and requirements of feature based software and system development in automotive industry?

To better understand challenges and the reasons for improvement expectations it is important to first gain understanding of how the system is developed and maintained today. Also, problems are identified, as obviously, the requirements of stakeholders deal with fixing them.

This section describes stakeholder expectation on more abstract level. Detailed goals and requirements to the new feature modeling approach are discussed in Section 5.3

5.2.1 Current Situation with respect to Product Line Engineering

As the results of the survey show, currently feature modeling, as well as product line engineering is not introduced in the studied organization at all. Transition from single systems to a systematic reuse approach is a part of the “New Generation Tool Chain” program (NGT). The main high level need for initiating this program was noted by the subsystem architect “because we offer different product variants to our customers”. Figure 6 shows the tools that are used today and the tools that were selected to improve the current processes in the future.

Development activities in GM correspond to the V-Model (Figure 7). New vehicle program starts from defining requirements and architecture for the future vehicle. On the bottom of the V-Model is creation of components that correspond to the requirements specification. These components are as general as possible to allow flexibility with respect to variations. Later in the development cycle the components are made specific to every vehicle that is produced. This process is called ‘calibration’. Calibration is done with the special GM tool and is based on the xml file that contains a list of Vehicle Option Codes (also called
as parameters) and their values. Vehicle Option Codes, and their possible values, are defined for each feature. For example, parameter TRANSMISSION_TYPE is used to specify transmission that is used in the vehicle. Two values are possible – manual and automatic. Number of parameters varies from component to component. Defining these parameters as a part of calibration process can be seen as binding of variability. Calibration process refers to product derivation (application engineering) in the Software Product Line terms.

![Figure 6. GM tool chain today and tomorrow](image)

Requirements are gathered at three levels of abstraction:

1. System level – customer requirements, not influenced by design or deployment
2. Domain level – decomposed from customer requirements, but still does not describe specific implementations or deployment
3. Component level – detailed design specifics of the component.

Nowadays, requirements specifications are done in MS Word. However, information about available features and their common and variable parts are not explicitly stated. As a result the system is hard to understand and hard to maintain.
Variability information is spread across use cases, specifications, calibration files and other artifacts. Most of the respondents could not state if variability information is captured or not and in what form. There is no specific system to describe variability of the features. They expressed that variants should be described, but not explicitly. Interviewees gave a number of answers on where and how variability information is captured today. Some of the answers were

- “…Ask people, as there is no specific system”
- “No feature models till now exist. Variability information in the form of xml files and calibration files”
- “Configuration files for ECUs and requirements specifications”

Within requirements specifications “interdependencies between different features and subsystems, and domains are modeled. However, dependencies on the parameter level are not captured currently.” The other interviewee added that “interdependencies are noted only sometimes”, because “we write our specifications in word documents the dependencies are sometimes lost. Sometimes there are links and references to some other feature, but they are not consistent”.

System evolution is a vital process that happens all the time. Most of the times a new feature is added or the existing one is modified. Rarely, the features are being deleted, as there is the need for support of older versions of the product. In GM change management process is well defined and commonly used. To edit any feature one can start a change request. This change request will be discussed within a special working group, approved by change review board and only after that it can be implemented. The review board consists of feature owners and architects. These people should decide and foresee any change impacts that are possible. This knowledge is not explicit and is based on human experience. As noted by the practitioners: “there is no direct system to get this knowledge, no traceability to another feature. It depends only on people” and “the impact analysis should be more transparent, but definitely not more complicated than it is now.” As a result a lot of requirements in the specification is not valid anymore and deleting of functionality is avoided. Also, the change should be made in several artifacts along the V-model: “We need to go through multiple documents along V-model and change them accordingly – to requirements, test cases, configuration data, calibration data. This is always done manually because the interdependencies are not explicitly available.” Despite these problems, the change management process worked well till now, this is confirmed by all the interviewees “It works pretty well. But in 1% of cases the mistake can still happen.”

Feature modeling is currently not in use, however the process of feature models introduction are already initiated. As one of the respondents told: “We are currently not using FM, but we are trying to get our development in this direction.” Trainings in basics of SPL and selected tool GEARS were organized. Nevertheless, the interviewees were still not confident about goals and benefits of FM introduction. Mostly, this confusion was caused by unavailability of familiar and realistic examples and guidelines, and also lack of experience in product line engineering: “…the concept of product line engineering is still not clear enough.”

The current process has some strength as some of the interviews showed. For example, one of the colleagues told that “The effort to set it up (to initiate the current process, author’s comment) is not very high, but you have a lot of effort to get information out of the system, because a lot of information is not available right now and you have to do a kind of reengineering. You go through a lot of documents, mostly WORD documents.” Strength is that “it works” and “it is easy to use it, as anyone can learn how to use a spreadsheet”. Otherwise, the methods used are mostly manual and system is hard to maintain.

The need for process improvement is highly understood by all the respondents. As the reasons for adopting PLE the following items were named:
- Need to know which variant are meant for which vehicle program
- Need to know what variants can be provided to customers
- Need to have an explicit information about product variability
- Need to work more efficiently
- Need to improve quality
- Need to increase availability of the information

That is what one of the respondents told: “For our team (System Engineering and Architecture) the benefit would be clear input information. Now we are struggling from time to time, we have to go to different people and ask for more information. Increased availability of information.”

But the main reason for all the improvement work is “handling the complexity”.

The problems that are faced using current approach will be discussed in the next section.

5.2.2 Problems with the Current Approach

The main problem noted by all the interviewees is high level of complexity of the system. Complexity arises from the huge amount of features and their variations, dependencies between them, lack of tool support.

Problems mentioned by interviewees are:

1. The process and variability handling depends on people. Most of the information should be asked from other departments or working group. As a result are communication overhead and a chance for misunderstandings. For example, “Currently we are doing this more or less by hand. For example, if you take a Park Assist. We have different variants of it. We are selecting those different variants manually. We need tool support here.”

2. There is no specific system to document variability. As already mentioned description of variants are done through requirements specification, spreadsheets and use cases. More time is required to understand and extract the needed information.

3. Information about feature variants is not available explicitly. Artifacts that describe variability today do not provide a structured view on differences between different variants. This information is mixed in with the specification of functions.

4. Need to look through a lot of text documents to extract information.

5. Interdependencies between features are not captured systematically (for example requires and excludes relationships).

6. No traceability between features and artifacts. For example, there is no way to trace a signal to the function it originates from. Also not enough traceability to perform impact analysis.

7. Artifacts are not consistent with each other. Sometimes links to other documents are defined, but not always.

8. The structure is not well defined. Information about features is not structured properly. For instance, there is no possibility to see the high level overview of features, their variants and interdependencies.

9. Component based development. There is a need to switch from component based development to feature based. Picking high level features (or even packages of
these features) is easier than picking components, as they are on the lower level of abstraction. Features should be traceable to components.

10. Poor understandability of existing notations to describe variability. This kind of information is in the text format. There is no graphical notation used.

These are the problems that were mentioned during the interviews by the stakeholders. The possible solution lies in creating feature models to structure features, define variable and common functionality and having information explicitly in one place. The problems are summarized in Table 5.

<table>
<thead>
<tr>
<th>Id</th>
<th>Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>The process is people dependent</td>
</tr>
<tr>
<td>P2</td>
<td>Variability information is spread across multiple artifacts</td>
</tr>
<tr>
<td>P3</td>
<td>Variability information is not explicit</td>
</tr>
<tr>
<td>P4</td>
<td>Need to extract variability information from multiple text documents</td>
</tr>
<tr>
<td>P5</td>
<td>Interdependencies between feature variants are not systematical</td>
</tr>
<tr>
<td>P6</td>
<td>No or bad traceability between variable features</td>
</tr>
<tr>
<td>P7</td>
<td>No or bad consistency between artifacts</td>
</tr>
<tr>
<td>P8</td>
<td>Variability information is not structured</td>
</tr>
<tr>
<td>P9</td>
<td>Component based development</td>
</tr>
<tr>
<td>P10</td>
<td>Poor understandability</td>
</tr>
<tr>
<td>P11</td>
<td>Impact analysis is experience based</td>
</tr>
<tr>
<td>P12</td>
<td>High level of complexity</td>
</tr>
</tbody>
</table>

Table 5 Problems with the current approach

The above mentioned problems are in causal relationship. The Id column from Table 5 indicates the problem identified and Figure 8 shows the cause/effect relationships. As seen, the central problem is low understandability that leads to high complexity and problems with impact analysis. P10 is influenced by variability information being not structured (P8), not explicit (P3), being spread across multiple artifacts (P2), bad consistency (P7) and also high complexity (P12). Because of bad consistency (P7) the variability information is not structured well (P8). And bad traceability (P6) leads to bad consistency (P7). As the process is people dependent (P1) the impact analysis is hard (P11). Of course, because the information is spread across different documents (P2) there is always the need to look through multiple text files (P4) that obviously is not efficient.

Figure 8 Problem Interdependencies
The solution of the above problems is seen in adopting product line engineering and introduction of tool support for variability management. However, this comes with additional challenges that are discussed in the next section.

5.2.3 Challenges in Adopting PLE

One part of the interview was dedicated to the challenges in creating and adopting new approach. The challenges mentioned during the interviews are discussed in this chapter.

- **Unavailability of modeling guidelines.** The new tool chain was selected by the headquarters; however the process and guidelines were not defined. Respondents spoke about confusion while starting modeling from structuring the models to integration of the models in the whole product lifecycle: "...we need some kind of definition on how to define variability".

- **High number of features.** What differs GM from other companies that adopted product line engineering methodologies is the number of features, variations, markets and so on. Moreover, the size is not big, but is constantly growing "It is growing even if you don’t plan it up. It might be the peculiarity of automotive. We tend to add more and more features and their variants". The approach provided should be helpful in dealing with this complexity.

- **Scalability.** One of the main problems for GM is scalability for the approach. There are different notations, tools and methodologies, but they should be well applicable for large scale product lines. As there is very high number of features the approach should allow modeling of these sizes and provide good usability during variability management. According to one of the interviewee “System is getting complex more and more, and this is our biggest task to cover its complexity.”

- **Granularity of models.** Another challenge is to define the right granularity of the models. To narrow down the models only variable information is needed. Practitioners told that they would like to model only key parameters, in order to make information manageable. This was told by one of the stakeholders: “Capture only the variability that is necessary. Not to model everything, but only relevant parameters. To have right size of the model to avoid additional complexity, but do not miss the necessary things”.

- **Evolvability.** The system is evolving constantly. It can be due to the faults found, change request or technology updates. “Change can happen during any time in development life cycle of the project. Changing should be supported.”

- **Support the complete development process.** Feature modeling should be supported through the whole development lifecycle. It is very important to integrate and trace variability through different artifacts. For example, the motivation for support of complete development process was given by one of the respondents: “For example, if I pick a feature like a Park Assist we should also consider the components of the vehicle where those features should be implemented and so on. Because we were planning to implement AUTOSAR and it gives us the possibility to implement our features on different components and that should be supported by the complete tool chain here.”

- **Keeping consistency and interdependencies.** As previously mentioned, the number of features is huge. The same holds for interdependencies between them. “Today the problem is that you have to read all the text in specification to understand the interdependencies”
5.2.4 Stakeholder Expectations to the Future Approach

In the last part of interview the interviewees were asked about their expectations and how they see the tool support and the variability management process in the future. The answers were high level, as the concrete requirements were not defined for them. The more detailed and structured goals were extracted from the workshops and will be presented in Section 5.3. The general view will be presented here.

First of all, stakeholders expect future approach to be compatible with GEARS (http://biglever.com) tool. Features should be modeled together with interdependencies and inclusion/exclusion logics. It is expected to have “...assertions for requires and excludes. Hierarchical inclusion of lower level FM to higher level FM. So we could form one integrated FM”. Also only variability should be modeled, as modeling of commonalities brings to much complexity.

As to the notations to represent feature models both graphical and structured text representation should be possible: “Graphical notation is good for small models, but text representation is needed for large models” or “I think that it should be able to have an overview of features in graphical format. It is more comfortable than the text format”. It was also said that there should be a possibility to “…switch between text and graphical notation”.

Created feature models should be used as a basis for instantiation of requirements specification, UML models, calibration parameters, test cases and release documentation. From the diagnostics perspective it would be nice to define a set of parameters for the feature diagnostics. Having feature models is expected to help with the impact analysis.

One of the interviewees shared his high level view on the tool possibilities that he would expect for feature modeling: “Currently we use Bill of Materials (BOM) tools to pick components. I think I would like something similar for features here. I think on the high level view it should be just possible to pick features and this should be transparent for the components in which feature will be implemented. I think what will be used for this is something like a graphical tool which will show a tree for a different features and feature variants. But on the high level prospective it would be fine to have a tool like our BOM tool, which will show the features which can be picked for the multiple vehicles.”

The functionality as graphical overview of features and interdependencies was mentioned by all the interviewees. Some of them noted the importance of switch between graphical and textual notations. Some pointed out that they would like to see model on different abstraction levels.

Good traceability is required. Traceability of commonalities is done through the introduced tools (see Figure 9). Requirements specification documents are traced to Design artifacts, Design artifacts are traced to Implementation artifacts, Implementation artifacts are traced to Test artifacts. However, variability is not traced properly with this approach. Figure 10 shows how Opel expects to trace variability between artifacts. The solution is to link created feature models to each of the artifact with the help of tools that are provided by BigLever GEARs.

![Figure 9 Traceability between common artifacts](image-url)
Expected solution should improve quality of the product, efficiency (time to market) and business understanding. In general, there should be a switch from manual (paper) work to the automated procedures using tools.

Figure 6 shows what tool changes are expected. First of all, requirements management is moved to IBM DOORS. That is a very important step that is not only beneficial for requirements engineering processes, but also because of possibility to use GEARS/DOORS bridge. As told during the interview “We are linked to DOORS here and we are planning to use Gears. The GEARS-DOORS link has to be done”. With the help of this bridge variation points based on feature models can be defined and traced between different artifacts. Secondly, parameter lists will be moved to GEARS feature models.

One of the important expectations is also possibility of configuration control “We would also need configuration management system to have a baseline on our features here. We also have to support parallel streams of development for features as well”.

It can be noted that most of the expectations and requirements are high level and concern only tool support. Following is the list of expectations:

- GEARS compatible
- Modeling of only variable features
- Modeling of excludes/requires interdependencies
- Hierarchical inclusion of lower level feature model into higher level feature model
- Graphical representation
- Structural/text representation
- Switch between different representations
- Automatic instantiation of requirements specification, documentation, test cases and so on
- Traceability links from feature model to artifacts
- DOORS/GEARS bridge usage
- Configuration control of feature models
- Parallel development of features

Next section discusses stakeholder goals and requirements in more details. The interview results will be used together with the workshop results to define list of measurable goals in order to help in choosing of the feature modeling approach.

5.3 Stakeholder Goals

The next step in finding of the improvement solution is specification of goals. Goals and their common understanding were obtained through interviews, workshop and meetings. At first, goals were defined using the ‘House of Quality’ [51] template, which is a common practice in the company. It was decided to use this format as it allows formalizing
expectation and requirements to the tools and the process used. Also this format was familiar to the people who work at the case organization. Afterwards goals were revised; prioritized and supporting examples were given.

However, Quality Function Deployment process is not enough, as our aim was to define some measurements that could be used to compare different strategies and also evaluate the created approach. To make goals measurable we also use Goal Question Metric (GQM) Paradigm.

Both QFD and GQM are goal oriented approaches, but they have a different focus. We used GQM to define a measurable program and a part of QFD to prioritize customer goals. In this context customers are the stakeholders of the feature models. The part of QFD template that was used is shown on Figure 11. Voices are gathered from different customers, from different points of view. So each stakeholder whose opinion is relevant in identifying requirements and priorities should list his/her voices in the left column of the table. Each requirement should be prioritized. Under the ‘Measures’ column technical or quality characteristics should be defined and the relation between Voices and ‘Measures’ are evaluated according to the scale. Taking into consideration these values Weighted important of each technical or quality characteristic is defined.

As already stated before, QFD was used as a basis to determine measurable goals that need to be met in order that the approach proposed satisfies the customer. The GQM approach [7] help to define and evaluate a set of goals using measurement. Figure 12 GQM Structure shows the components of this model. GQM graph determines measurable goal (or goals), number of questions that are linked to the goal and metrics used to measure them. Interpretation model is used to understand the results of the measurement.
Section 5.3.1 describes customer voices obtained by QFD template. Derived GQM is discussed in Section 5.3.2.

![GQM Structure](image)

Figure 12 GQM Structure [76]

5.3.1 Improvement Goals. QFD

Table 6 shows the filled in QFD template. Customer voices were gathered from the perspective of three stakeholder roles, namely:

- Requirements Engineer
- Validation Engineer
- Release Engineer

The voices of Requirements Engineer are:

- Wants to unambiguously and correctly specify product variants and their associated requirements. The task of requirements engineer is to correctly identify and describe requirements of the features and their variants. Configurations of requirements have to be specified correctly as well. This also means that there should be a way to specify needed types of dependencies between features.

- Wants an easy way to specify variation in the requirements. ‘Easy way’ is having ambiguous meaning. Here we refer to such a way of specification that a person can learn how to do it after training. The approach should be supported by the tool. An easy way can also mean that the way of specifying requirements and variants are precisely defined. Everybody knows and can understand the process of adding variation to the requirement.

- Wants a uniform way to specify variation in the requirements. The property of uniformity refers to the standard specified process. It should be defined how to address similar common situations during modeling. This contributes to better understandability and reusability of the variable information. This also should be more efficient, as the solutions for the most common problems are already defined, and people do not have to spend time for trying something out and then rework.

The voices of Validation Engineer and Release Engineer are:

- Wants an easy way to understand variation in the requirements. It is very important to be able to understand fast the variations in the requirements and dependencies between them.

- Wants to have correct and complete information regarding product variants and their associated requirements. As a user of created feature models, validation and release engineers need means to get information about product variants. This information should be complete and correct to avoid missing out important configurations or fault propagation.
- Wants the variation information in a uniform way. When the same problems are addressed similarly and in the standardized by the company way, there is a better chance that interpretation of the models will be common.

As seen from the Table 6 the expectations for these stakeholders are the same. The reason for that is that they are responsible for using the variability information, in contrast to Requirements Engineer, who is the one responsible for adding the variability information.

Some of the requirements are the same for all the stakeholders. For example, the requirement of having a local impact of change is valid for all three stakeholders. This requirement means that change should influence as less other modules as possible. Of course, if something should be changed in the model all the stakeholders of this feature model want to change it only once. This requirement contributes a lot to satisfying criteria of feature model modularity. The second common requirement is “Wants to avoid unnecessary complexity”. To understand what was meant by this statement it is first need to define “complexity”. Afterwards, specify “necessary” and “unnecessary” complexity.
Also, quality criteria that need to be improved were defined. During this work we concentrated on the following quality criteria:

- Correctness
- Completeness
- Modularity
- Understandability
- Maintainability
- Consistency (Traceability)
- Reusability

### Table 6 Customer Voices

<table>
<thead>
<tr>
<th>Customer Grouping</th>
<th>Voices</th>
<th>Item No.</th>
<th>Importance</th>
<th>Understandability</th>
<th>Maintainability</th>
<th>Modularity</th>
<th>Correctness</th>
<th>Completeness</th>
<th>Consistency (Traceability)</th>
<th>Reusability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements Engineer</td>
<td>1</td>
<td>Wants to unambiguously and correctly specify product variants and their associated requirements</td>
<td>1</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>9</td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Wants an easy way to specify variation in the requirements</td>
<td>2</td>
<td>2</td>
<td>9</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Wants to have a local impact of changes</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>9</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Wants to avoid unnecessary complexity</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>9</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Wants a uniform way to specify variation in the requirements</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td>Validation Engineer</td>
<td>2</td>
<td>Wants an easy way to understand variation in the requirements</td>
<td>6</td>
<td>3</td>
<td>9</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Wants to have a local impact of changes</td>
<td>7</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>9</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Wants to have correct and complete information regarding product variants and their associated requirements</td>
<td>8</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>9</td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td>Release Engineer</td>
<td>2</td>
<td>Wants to avoid unnecessary complexity</td>
<td>9</td>
<td>2</td>
<td>1</td>
<td>9</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Wants the variation information in a uniform way</td>
<td>10</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Wants an easy way to understand product variants</td>
<td>11</td>
<td>2</td>
<td>9</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Wants to have a local impact of changes</td>
<td>12</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>9</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Wants to have correct and complete information regarding product variants and their associated requirements</td>
<td>13</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>9</td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Wants to avoid unnecessary complexity</td>
<td>14</td>
<td>3</td>
<td>1</td>
<td>9</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Wants the variation information in a uniform way</td>
<td>15</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>9</td>
</tr>
</tbody>
</table>

Company Ratings

- Strong - 9
- Medium-3
- Weak - 1

<table>
<thead>
<tr>
<th>Relationship Strength</th>
<th>Targets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Good</td>
<td>5</td>
</tr>
<tr>
<td>Medium</td>
<td>4</td>
</tr>
<tr>
<td>Weak</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Weighted Importance</th>
<th>% Importance</th>
</tr>
</thead>
<tbody>
<tr>
<td>14.3</td>
<td>14.3</td>
</tr>
<tr>
<td>14.5</td>
<td>14.5</td>
</tr>
<tr>
<td>16.7</td>
<td>16.7</td>
</tr>
<tr>
<td>16.5</td>
<td>16.5</td>
</tr>
<tr>
<td>13.5</td>
<td>13.5</td>
</tr>
<tr>
<td>12.5</td>
<td>12.5</td>
</tr>
</tbody>
</table>
Customer voices contribute one or another quality criterion, as seen from the Table 6. So, for instance, requirement No.1 is strongly related to Correctness and Completeness. Requirement No.2 contributes to Understandability. Based on the customer voices the prioritized list of quality goals are following:

1. Correctness
2. Completeness
3. Modularity
4. Understandability
5. Maintainability
6. Consistency (traceability)
7. Reusability

Prioritizing quality criteria is extremely important during decision making in favor of one or another strategy or approach. For example, when two approaches are beneficial for one criterion and decreases the quality of another the priorities can be used. In such a way in case of tradeoffs, the reasonable choice can be made.

To better understand what is meant by mentioned quality criteria in the context of this project refer to Table 7. Consequences of not fulfilling a certain quality are also listed in Table 7.
Quality | Explanation | Consequence
--- | --- | ---
Correctness | Model should not contradict reality. Reality here refers to requirements specification. Created Product Configurations should be a valid product. | If the model is not correct than all other artifacts will contain faults, that lead to incorrect system.
Completeness | All the variants of the feature should be modeled. All the dependencies between features should be modeled. Also it should be possible to define all valid configurations. | If a feature is missing a test cannot detect a defect. The same holds for incompleteness of dependencies.
Modularity | Feature Model should be ideally divided into a set of modules. The criteria for division could be different from semantics to number of dependencies. | Structural complexity of feature model in GM is too high to have a monolithic model. Failing to come up with a good modular structure will lead to bad maintainability and understandability.
Understandability | Introduction of Feature Model should contribute to understanding of variability of the system by giving an overview of the features, variants and dependencies. Considering Understandability as a property of feature model it means representing of variability information in the way easiest to process | Requirements are not implemented correctly. Inappropriate design is created. The design is too complex. Validation is harder to execute that leads to missing out defects or to taking valid functionality as a defect.
Maintainability | Evolution of feature models cannot be avoided. Though, the initial modeling should be done in the way that makes further maintaining manageable. The aspects of concern can be: no (or minimal) duplication of information, separation of concerns and so on. Evolution is further discussed in Section 5.4. | Duplicate information leads to inconsistency. Analyses and execution of change requests are difficult, as the person should make change in all the occurrences of the change.
Consistency | Traceability of variable information between artifacts, between elements of model in different levels of hierarchy | Missing out features.
Reusability | Reusability here is used in the sense of avoiding redundant and duplicated information. Features (or parts of feature models) should not be copied, but either used by reference. | Increases maintainability and understandability difficulty.

Table 7 Quality Goals Explained

5.3.2 Improvement Goals. GQM

This chapter will explain the derivation of goals according to GQM. The overall goal obtained from QFD can be defined as follows: to analyze process of managing variability with the purpose of improving satisfaction of Requirements Engineer (RE), Validation Engineer (VE), Release Engineer (RelE). This goal is very high level and abstract, that is why it is divided into seven smaller goals. In general these goals focus on improvement of quality attributes stated in the previous chapter. To be more specific and to derive better
questions each goal is further decomposed into subgoals. Subgoals are formulated in the form of requirements that should be fulfilled. Then questions are stated, metrics and interpretation model are given to answer them. In this work GQM is used to describe measures that allow assessing quality attributes. Quality attributes represent customer voices. So achievement of GQM goals leads to improvements of quality attributes and thus to satisfaction of customer voices.

The GQM spreadsheet is used to compare different approaches to feature modeling, to state if the goals are fulfilled under the mentioned guideline and to compare if a new guideline (or approach) is better than the old one in the future. The full measurement plan is not provided in this thesis, as it was not feasible and also because the focus of the work is different.

The first goal G1 is to improve understandability of the system with respect to variability information. Table 8 gives a definition of a goal according to GQM template. Table 9 lists subgoals, questions and metrics for evaluation. S1 says that specification of product variant should be done easier than before. S1 deals with understandability during specification of variants. Questions Q1/Q2 do not aim directly at understandability but on efficiency/effort. Nevertheless, it is assumed that increased efficiency goes hand in hand with improved understandability. Therefore a bad experts' rating for this category would leave doubt on the understandability of the (GEARS' adapted) approach. The metric for Q1 is expert based. Experts should be asked about their rating of Q1 on a Likert scale of "Strongly Agree", "Agree", "Undecided / Don't know", "Disagree", "Strongly Disagree". Survey forms can be created to gather required information. To answer question Q2 time needed to specify feature and its variants can be measured. Gathered time should be compared with the baseline time. If less time is required to add features and variants then the approach is more understandable.

| G1. Evaluate the process and results of feature modeling for the purpose of improvement with respect to understandability from the point of view of family engineer |
|---|---|---|

Table 8 GQM definition of G1

Subgoal S2 defines understandability related to usage of already specified feature variants. How easily and fast product variants can be understood is a subjective measure. However, some obvious objective criteria can be found to make it more explicit. Questions Q3/Q4/Q5 focus on speed of understanding and amount of artifacts or documents that should be looked through to obtain the information. It is assumed that structured documents are easier to understand. Also it is easier to obtain information if it is gathered in one source, but not spread among multiple artifacts. Also, if you understand information faster than before, then it means that it is easier.

S3 states that overall understandability will benefit if there is an option to get an overview of high level features and their variants. To be able to give answer the number of available abstraction views should be counted. This goal should be fulfilled by the tool selected for usage. Subgoal S4 contributed to better understandability by requesting a property of concerns separation. Mechanism of views for different roles and stakeholders can decrease cognitive complexity, thus improves understandability.

Understandability is also influenced by how dependencies between features can be analyzed (see subgoal S5). Question Q8 aims to quantify understandability of dependencies in terms of effort. Increased efficiency in understanding relations between features improves understandability of the system as a whole. To measure it an experiment can be set up, that measures time needed to understand dependencies between features. Also to determine if S5 is fulfilled we need to know if approach describes types of dependencies and how they
should be expressed (Q9). When this information is defined in the guideline then users are able to identify dependency types and thus comprehend better. The evaluation is done by experts, who should revise the guideline and determine if all needed dependencies are described in the good way.

<table>
<thead>
<tr>
<th>Subgoal (Requirements)</th>
<th>Questions</th>
<th>Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1. Specification of product variants are done easier than before</td>
<td>Q1. Is Specification of product variants done easier than before?</td>
<td>Expert based</td>
</tr>
<tr>
<td></td>
<td>Q2. How efficient is specification of product variability?</td>
<td>Time needed to specify feature and its variants</td>
</tr>
<tr>
<td></td>
<td>Q3. In what artifacts is variability information described?</td>
<td>Types of Artifacts</td>
</tr>
<tr>
<td></td>
<td>Q4. In how many artifacts is variability described?</td>
<td>Number of Artifacts</td>
</tr>
<tr>
<td></td>
<td>Q5. What is the time required to understand product variants?</td>
<td>Time required to understand product variants</td>
</tr>
<tr>
<td>S2. Understanding of product variants are done easier than before</td>
<td>Q6. Can features and variations be viewed at different level of abstraction?</td>
<td>Number of different levels of abstraction available for viewing</td>
</tr>
<tr>
<td></td>
<td>Q7. Can different views of the feature model be defined?</td>
<td>Number of different views</td>
</tr>
<tr>
<td></td>
<td>Q8. How much time is needed to understand dependencies between features?</td>
<td>Time to understand dependencies</td>
</tr>
<tr>
<td></td>
<td>Q9. Does the approach describe possible dependencies between features and the way to express them?</td>
<td>Expert based</td>
</tr>
</tbody>
</table>

Table 9 Subgoals, Questions and Metrics for G1 (Understandability)

Goal G2 is to improve correctness of the feature model with respect to variability information. Table 10 gives definition of G2 according to GQM. Derived subgoals, questions and metrics are defined in Table 11. Subgoals S6, S7, S8 refine the meaning of correctness in this particular context. To answer questions Q10, Q11, Q12, Q13 inspections and review of models, product configurations and requirement specifications should be done by experts. Variability management approach cannot ensure that created model is correct, as the activities are human based. However, restrictions can be posed; checklists can be provided to avoid common mistakes.

| G2. Evaluate the process and results of feature modeling for the purpose of Improvement with respect to correctness of variability information from the point of view family engineer | |

Table 10 GQM definition of goal G2
### Subgoals (Requirement) | Questions | Metrics
--- | --- | ---
S6. Model should not contradict requirement specification (reality) | Q10. Does feature model contradict to the requirement specification documents? | Number of features that contradict specification

S7. Created Product Configurations should be a valid product | Q11. Can all valid products be described? | Number of valid products to Number of Defined configurations

Q12. Is it possible to define a configuration that contradicts specification? | Expert based

S8. Specification of requirement variants is unambiguous (There is only 1 way to interpret variability in FM) | Q13. Is naming of features unambiguous? | Expert based

### Table 11 Subgoals, Questions and Metrics for G2 (Correctness)

One of the important problems noted is traceability between different artifacts, descriptions of features on the different abstraction levels. Goal G3 (Table 12) defines the property of consistency of variability information that should be fulfilled by the selected feature modeling approach. Refinements of the G3 to requirements S9, S10, and S11 tend to make sure that variability information is traceable between artifacts and levels of abstraction (Table 13). Questions Q14 – Q17 should be answered by experts by evaluating the approach and the selected tools. To measure Q18 number of identified patterns should be compared to number of different solutions found. This can be achieved by review artifact base by quality department.

| G3. Evaluate the Process and results of feature modeling for the purpose of improvement with respect to consistency of variable information from the point of view family engineer |
|---|---|---|

### Table 12 GQM definition of goal G3

| Subgoal (Requirements) | Questions | Metrics |
--- | --- | ---
S9. Variable information should be traceable between different artifacts | Q14. Can the variability information be traceable between different artifacts? | Expert based

Q15. Is traceability ensured by tool support? | Expert based

S10. Variable information should be traceable between different levels of abstraction | Q16. Can the variability information be traceable between different levels of abstraction? | Expert based

S11. Variability information should be described in the uniform way | Q17. Is the single notation used? | Expert based

Q18. Are similar situations addressed similarly? | Number of identified patterns to number of different solutions

### Table 13 Subgoals, Questions and Metrics for G3 (Consistency)

Goal G4 (Table 14) explores the quality of completeness. Completeness depends both on the notation capabilities, selected structuring method, tool support and methodological guidance provided. In this specific context completeness are examined from the two prospective (S12, S13 in Table 15):

- Modeling all variants of the feature
Modeling all the dependencies between features

Moreover, we should divide between the possibility to model and if everything was actually modeled. The first aspect should be ensured by approach creation (by construction); the second aspect is ensured by providing quality reviews of modeling results. Q19 and Q20 tend to answer if all the variants from requirement specification were modeled by counting the number of feature variant in both artifacts. To answer Q21 experts first should define kinds of dependencies that are required. Afterwards, this information should be compared with the dependencies that can be technically described.

<table>
<thead>
<tr>
<th>Subgoal (Requirements)</th>
<th>Questions</th>
<th>Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>S12. All possible variants of the feature should be modeled</td>
<td>Q19. How many feature variants are described in Requirements Specification?</td>
<td>number of feature variants in requirements specification</td>
</tr>
<tr>
<td></td>
<td>Q20. How many feature variants are modeled?</td>
<td>number of feature variants in feature model</td>
</tr>
<tr>
<td>S13. All dependencies between features should be modeled</td>
<td>Q21. Is it possible to model all the dependencies?</td>
<td>Types of dependencies that should be modeled VS types of dependencies that are possible to model with the selected approach</td>
</tr>
<tr>
<td></td>
<td>Q22. Are all the dependencies between features modeled?</td>
<td>Expert based</td>
</tr>
</tbody>
</table>

Table 14 GQM definition of goal G4

S14 does not directly contribute to improving of modularity (Table 16 and Table 17). However, the prospect of change impact is an important angle of modular system. To evaluate the locality of change three metrics should be gathered: number of affected features, assertion and dependencies; number of affected artifacts; number of affected models. Of course the less the number the better is modularity of the created models and artifacts as a whole.

<table>
<thead>
<tr>
<th>Subgoal (Requirements)</th>
<th>Questions</th>
<th>Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>S14. Minimize impact of the change</td>
<td>Q23. Is impact of the change local?</td>
<td>Number of affected features, assertions, dependencies</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Number of affected artifacts</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Number of affected models</td>
</tr>
</tbody>
</table>

Table 17 Subgoals, Questions and Metrics for G5 (Modularity)

As to the decreasing of redundancy in variability information (G6 in Table 18) the important aspects are (Table 20) uniformity of the information representation (S15) and no duplicating of information (S16). S16 is a property of the technical abilities of the tool and
the identified approach. The notation should allow access of information by reference. To determine if the redundancy level is decreased, the number of copied parts of feature model should be compared.

| G6. Evaluate the Feature modeling process and results for the purpose of Decreasing redundancy with respect to Variability information from the point of view Family engineer |
|---|---|---|---|

Table 18 GQM definition of goal G6

It is assumed that subgoals S17-S20 (Table 20) contribute to improving of maintainability (Table 19). S17 and S18 were discussed previously as they correspond to S11 and S16 respectively. However, Q25 and Q26 aim to define the patterns where complexity can be reduced by applying some structural solution. Moreover, maintainability highly depends on work separation capabilities. If the parts of the model can be maintained in parallel than the approach (together with its technical realization) is more efficient in the aspect of maintainability.

| G7. Evaluate the System artifact base for the purpose of improving with respect to maintainability from the point of view Family engineer |
|---|---|---|---|

Table 19 GQM definition of goal G7

<table>
<thead>
<tr>
<th>Subgoal (Requirements)</th>
<th>Questions</th>
<th>Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>G7</td>
<td>S17. (S11) Variability information is specified in the unified way S18. (S16) Variability information should not be duplicated S19. Unnecessary complexity should be avoided</td>
<td>Q25. Can the patterns of unnecessary complexity be defined?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Q26. Are the defined complexities avoided?</td>
</tr>
<tr>
<td></td>
<td>S20. Separation of work should be allowed</td>
<td>Q27. Can model be used and developed in parallel?</td>
</tr>
</tbody>
</table>

Table 20 Subgoals, Questions and Metrics for G6 (Redundancy) and G7 (Maintainability)

This section gave overview of the goals defined by following GQM paradigm. These goals are refined in the number of subgoals, namely, requirements that should be fulfilled in order to satisfy the main goals. To help experts to evaluate the fulfillment of the requirements the questions/metrics tables are given. This template should be used for two purposes:

1. To decide between different approaches or methods (by answering only those question that are applicable)
2. To decide, in the future, if the new approach (guideline) to feature modeling is better than the currently used one.

In this thesis these requirements are used mostly for the first purpose. It is obvious, that applying template to compare between the old and the current approach in the context of the thesis is impossible, because there is no approach defined.

5.4 Evolution Scenario

During the workshop the possible changes to the feature model were discussed. This evolution scenario was created by practitioner from the case organization and the researcher. Evolution was deemed as an important issue in the creation of feature modeling approach. Writing down most probable actions that can be taken towards feature models will help in evaluation of approaches and structures selection.

Evolution of the software systems is inevitable. Changes to feature models can happen for several reasons: internal reorganization, changes in functionality, extension of functionality or reduction of functionality [29]. Because of these reasons the following evolution scenario that covers all reorganizations is possible:

1. Create model.
2. Create dependencies.
3. Add leaf element.
4. Add new feature group.
5. Remove feature with dependencies. This action is almost not possible, as it is hard to predict the impact of functionality deletion.
6. Remove feature group. This action is possible only in the context of internal reorganization when the same feature group is later created somewhere else in the hierarchy.
7. Change level of hierarchy of the feature.
8. Add constraint.
11. Performance upgrade. The old version is often still available, but the newer version (that is mostly more expensive) are already sold in the new market, or the upgrade is just being planned for next vehicle program. For example, VW sells old architecture car to Asian countries, and newer technology for EU market.

Possibility to carry out described actions, and also how easily it can be done should be taken in consideration during construction of the approach.

5.5 Summary

This chapter discussed the state of the practice in variability management, and feature modeling in particular, based on the case study in automotive company. The data collection methods used were semi structured interviews, workshops and documentation review. During this phase of the case study the aim was to answer the first research question: “What are the challenges and requirements of feature based software and system development in automotive industry?” We summarized the current issues in the organization, challenges in adopting product line approach and the requirements of stakeholders to the future feature modeling approach.
Table 21 shows the problems that were discovered in the case organization with respect to variability management. Fixing these problems is the first set of requirements. The respective goals are summarized in the right column of the table.

<table>
<thead>
<tr>
<th>Problem (Anti goal)</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1. The process is people dependent</td>
<td>R1. Process is supported by tool</td>
</tr>
<tr>
<td>P2. Variability information is spread across multiple artifacts</td>
<td>R2. Variability information is captured in one place</td>
</tr>
<tr>
<td>P3. Variability information is not explicit</td>
<td>R3. Explicit modeling of variability by feature modeling</td>
</tr>
<tr>
<td>P4. Need to extract variability information from multiple text documents</td>
<td>R4. Variability information is captured in one place and in structured form</td>
</tr>
<tr>
<td>P5. Interdependencies between feature variants are not systematical</td>
<td>R5. Interdependencies between features are captured systematically</td>
</tr>
<tr>
<td>P6. No or bad traceability between variable features</td>
<td>R6. Features can be traceable</td>
</tr>
<tr>
<td>P7. No or bad consistency between artifacts</td>
<td>R7. Artifacts are consistent to each other</td>
</tr>
<tr>
<td>P8. Variability information is not structured</td>
<td>R8. Variability information is structured</td>
</tr>
<tr>
<td>P10. Poor understandability</td>
<td>R10. Easy to understand</td>
</tr>
<tr>
<td>P11. Impact analysis is experience based</td>
<td>R11. Impact analysis is more automatic</td>
</tr>
<tr>
<td>P12. High level of complexity</td>
<td>R12. Reduce of complexity</td>
</tr>
</tbody>
</table>

Table 21 Problems with the current approach and respective improvement goals

The answer to the first part of research question about challenges of automotive industry is summarized in the left column of Table 22. So, the next list of requirements can be extracted from the challenges and is summarized in the right column of Table 22. These requirements deal with feature modeling approach, but not with a tool support.

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1. Unavailability of modeling guidelines</td>
<td>R13. Modeling guidelines should be available for the approach</td>
</tr>
<tr>
<td>C2. High number of features</td>
<td>R14. Approach should be scalable</td>
</tr>
<tr>
<td>C3. Scalability</td>
<td>R14. Approach should be scalable</td>
</tr>
<tr>
<td>C4. Granularity of models</td>
<td>R15. Approach should give advice on granularity of models</td>
</tr>
<tr>
<td>C5. Evolvability</td>
<td>R16. Approach should support evolution</td>
</tr>
<tr>
<td>C6. Support the complete development process</td>
<td>R17. Feature modeling approach should be incorporated with the whole development process</td>
</tr>
<tr>
<td>C7. Keeping consistency and interdependencies</td>
<td>R18. Approach should define and give support for consistency and keeping interdependencies</td>
</tr>
</tbody>
</table>

Table 22 Challenges of feature modeling adoption in the case organization and respective improvement goals
Table 23 summarizes those requirements to the feature modeling approach, tool support and process in general that were directly stated by the stakeholders during interviews.

<table>
<thead>
<tr>
<th>Requirements</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>R19. GEARS compatible</td>
<td></td>
</tr>
<tr>
<td>R20. Modeling of only variable features</td>
<td></td>
</tr>
<tr>
<td>R21. Modeling of excludes/requires interdependencies</td>
<td></td>
</tr>
<tr>
<td>R22. Hierarchical inclusion of lower level feature model into higher level feature model</td>
<td></td>
</tr>
<tr>
<td>R23. Graphical representation</td>
<td></td>
</tr>
<tr>
<td>R24. Structural/text representation</td>
<td></td>
</tr>
<tr>
<td>R25. Switch between different representations</td>
<td></td>
</tr>
<tr>
<td>R26. Automatic instantiation of requirements specification, documentation, test cases and so on</td>
<td></td>
</tr>
<tr>
<td>R27. Traceability links from feature model to artifacts</td>
<td></td>
</tr>
<tr>
<td>R28. DOORS/GEARS bridge usage</td>
<td></td>
</tr>
<tr>
<td>R29. Configuration control of feature models</td>
<td></td>
</tr>
<tr>
<td>R30. Parallel development of features</td>
<td></td>
</tr>
</tbody>
</table>

Table 23 Explicitly Stated Requirements to FM

The fourth group of requirements is represented as customer voices (QFD) and measurable goals (GQM). The quality attributes are also defined and are important in satisfaction of customer voices. The Table puts together quality requirements and related refined functional requirements. Metrics are not presented in the summary, but can be reviewed in Section 5.3.

<table>
<thead>
<tr>
<th>Quality Requirements</th>
<th>Refinements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correctness</td>
<td>R31. Model should not contradict requirement specification(reality)</td>
</tr>
<tr>
<td></td>
<td>R32. Created Product Configurations should be a valid product</td>
</tr>
<tr>
<td></td>
<td>R33. Specification of requirement variants is unambiguous (There is only 1 way to interpret variability in FM)</td>
</tr>
<tr>
<td>Completeness</td>
<td>R34. All possible variants of the feature should be modeled</td>
</tr>
<tr>
<td></td>
<td>R35. All dependencies between features should be modeled</td>
</tr>
<tr>
<td>Modularity</td>
<td>R36. Minimize impact of the change</td>
</tr>
<tr>
<td>Understandability</td>
<td>R37. Specification of product variants are done easier than before</td>
</tr>
<tr>
<td></td>
<td>R38. Understanding of product variants are done easier than before</td>
</tr>
<tr>
<td></td>
<td>R39. Comprehensive overview of the high level features and their variants</td>
</tr>
<tr>
<td></td>
<td>R40. Defined view according to point of interest</td>
</tr>
<tr>
<td></td>
<td>R41. Dependencies between features and their variants can be analyzed</td>
</tr>
<tr>
<td>Maintainability</td>
<td>R42. Variability information is specified in the unified way</td>
</tr>
</tbody>
</table>
Variability information should not be duplicated
Unnecessary complexity should be avoided
Separation of work should be allowed

Consistency (traceability)
Variable information should be traceable between different artifacts
Variable information should be traceable between different levels of abstraction
Variability information should be described in the uniform way

Reusability
Variability information is specified in the unified way
Variability information should not be duplicated

Table 24 Quality Goals Summarized

The four requirements groups have duplicates or issues that are very similar to each other. To create the final list of requirements/goals these issues were removed. Also GEARS specific requirements were removed as they are already included in the context of this work. Table 25 below represents the requirements set to the process of feature modeling, tool support and feature modeling approach in general. The left column of the table shows the reference to the paper where the similar requirement was identified. “X” shows that no correspondence was found. This mapping is preliminary and does not mean that there are no other papers discussing the same requirements. The material used here is based on the set of papers retrieved during literature review in the next section. Some of the requirements found in this study could not be directly mapped to some previous work. However, they could be addressed by the proposed approaches.

<table>
<thead>
<tr>
<th>Case Study Requirements</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Process is supported by tool</td>
<td>[32][39]</td>
</tr>
<tr>
<td>2. Variability information is captured in one place</td>
<td>X</td>
</tr>
<tr>
<td>3. Explicit modeling of variability by feature modeling</td>
<td>[31]</td>
</tr>
<tr>
<td>4. Variability information is structured</td>
<td>X</td>
</tr>
<tr>
<td>5. Feature based development</td>
<td>X</td>
</tr>
<tr>
<td>6. Impact analysis is more automatic</td>
<td>[31]</td>
</tr>
<tr>
<td>7. Modeling guidelines should be available for the approach</td>
<td>[39]</td>
</tr>
<tr>
<td>8. Approach should be scalable</td>
<td>[32]</td>
</tr>
<tr>
<td>9. Approach should give advice on granularity of models</td>
<td>X</td>
</tr>
<tr>
<td>10. Approach should support evolution</td>
<td>[32]</td>
</tr>
<tr>
<td>11. Feature modeling approach should be incorporated with the whole development process</td>
<td>[32]</td>
</tr>
<tr>
<td>12. Modeling of only variable features</td>
<td>X</td>
</tr>
<tr>
<td>13. Modeling of excludes/requires interdependencies</td>
<td>[31]</td>
</tr>
<tr>
<td>14. Hierarchical inclusion of lower level feature model into higher level feature model</td>
<td>X</td>
</tr>
<tr>
<td>15. Graphical representation</td>
<td>[32]</td>
</tr>
<tr>
<td>16. Structural/text representation</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>17.</td>
<td>Switch between different representations</td>
</tr>
<tr>
<td>18.</td>
<td>Automatic instantiation of requirements specification, documentation, test cases and so on</td>
</tr>
<tr>
<td>19.</td>
<td>Configuration control of feature models</td>
</tr>
<tr>
<td>20.</td>
<td>Parallel development of features</td>
</tr>
<tr>
<td>21.</td>
<td>Model should not contradict requirement specification (reality)</td>
</tr>
<tr>
<td>22.</td>
<td>Created Product Configurations should be a valid product</td>
</tr>
<tr>
<td>23.</td>
<td>Specification of requirement variants is unambiguous (There is only 1 way to interpret variability in FM)</td>
</tr>
<tr>
<td>24.</td>
<td>All possible variants of the feature should be modeled</td>
</tr>
<tr>
<td>25.</td>
<td>All dependencies between features should be modeled</td>
</tr>
<tr>
<td>26.</td>
<td>Minimize impact of the change</td>
</tr>
<tr>
<td>27.</td>
<td>Specification of product variants are done easier than before</td>
</tr>
<tr>
<td>28.</td>
<td>Understanding of product variants are done easier than before</td>
</tr>
<tr>
<td>29.</td>
<td>Comprehensive overview of the high level features and their variants</td>
</tr>
<tr>
<td>30.</td>
<td>Defined view according to point of interest</td>
</tr>
<tr>
<td>31.</td>
<td>Dependencies between features and their variants can be analyzed</td>
</tr>
<tr>
<td>32.</td>
<td>Variability information is specified in the unified way</td>
</tr>
<tr>
<td>33.</td>
<td>Variability information should not be duplicated</td>
</tr>
<tr>
<td>34.</td>
<td>Unnecessary complexity should be avoided</td>
</tr>
<tr>
<td>35.</td>
<td>Separation of work should be allowed</td>
</tr>
<tr>
<td>36.</td>
<td>Variable information should be traceable between different artifacts</td>
</tr>
<tr>
<td>37.</td>
<td>Variable information should be traceable between different levels of abstraction</td>
</tr>
</tbody>
</table>

Table 25 Requirements to the VM modeling approach in the case organization

This list is used in the next sections to evaluate the approach or a set of approaches (the combination of approaches in order to achieve the goals). During selection, or synthesizing, of feature modeling approach this list is used to decide on the suitability for the case organization. Moreover, the defined GQM goals will be used to compare current situation with the situation after the approach introduction.
6 STATE OF THE ART

This chapter presents results of state-of-the-art analysis in the subject of feature modeling. The main goal was to systematically gather information about feature modeling techniques, approaches and structures. As a matter of fact, feature modeling by itself is a very broad field. The goal was to find research papers that provide solutions to the problems identified in the case organization. Papers reporting case studies, methodological guidelines and new ideas that are applicable in automotive industry were of a special interest. The idea was to find if there is any approach that is good enough to adopt by the case organization. The selected approach should satisfy the goals and requirements specified in Section 5.3.2. The selected approach should also be technology compatible with GEARS tool. However, during the literature review we did not limit ourselves to technology level, but tried to find technology independent solution.

Next sections describe the process of information retrieval, inclusion criteria for papers, critical evaluation of found feature modeling techniques. During this literature study, statistical analysis of retrieved articles was not that important. Some numbers were found in the previously made reviews. The main focus is the qualitative data and research ideas.

6.1 Research Method

As previously mentioned in the Research Methodology chapter of this thesis, literature review will be performed systematically by using the “snowball” technique [28]. The original guidelines of Kitchenham will not be followed, as we want to concentrate on the extracted information and possible solutions, and not on the aggregation of evidence.

Checking references of the relevant articles gives a broader overview of relevant literature, provides with the primary sources of ideas. Checking who referenced the article later (check forward) speaks about paper importance and the further development of the idea. This kind of search helps to find any systematic literature reviews, case studies, or comparative evaluation of the ideas. The main benefit of using snowballing for this work was retrieving the primary studies from literature reviews and surveys.

The requirements to the final approach elicited in the previous chapter are based on the case organization; however they can be applied to the other automotive companies.

So, this process will be followed.

1. Identification of key words and phrases
2. Combining search strings
3. Search and select articles
4. Check references backward
5. Check references forward
6. Identify relevance
7. Document
8. Make notes

Literature review is done to give an answer for research question 2: “What hierarchical feature modeling approaches exist in literature, what are their characteristics and how are they applied in industry?”

Following sections will address the process of search, selection and extracting information from papers in more detail.
6.1.1 Search

The search was done electronically through the main journals, citation databases and conference proceeding, that publish papers in the fields of software product line, variability management, feature modeling or automotive software engineering. Electronic databases that were used to identify papers were:

- Engineering Village – provides single interface to INSPEC and Compendex databases.
- IEEEXplore – publisher of important conference proceedings.
- ACM Digital Library – publisher of important conference proceedings and journals in software engineering.
- Google Scholar – searches over the whole electronically available papers.
- ScienceDirect – publisher of main journals in computer science.

Through the search in these digital libraries relevant papers were retrieved. Articles mostly belong to one of the following journal or conference proceeding:

- **Software Product Line Conference Proceedings** are one of the main resources of information, as it yearly unites together researchers and practitioners in software product lines. It covers the topics of variability management and feature modeling as well.
- **Variability Modeling of Software Intensive Systems Workshop** concentrates on variability modeling and management issues through the whole life cycle of systems.
- **Automotive Requirements Engineering Conference** provides papers on specific problems of requirements management in automotive industry.

The search process was done iteratively through the whole time (see Figure 13). First, a set of keywords (Table 26) from research question was defined. Then search strings were created by combining keywords group by AND operator. Several search strings were created to retrieve information on different aspects (e.g. guides in automotive domain, and guides independent of domain). Search strings were used in the digital library search. As a result a number of papers were retrieved. After that papers’ references were checked. As a result the keywords were updated and the search was repeated from the beginning.

<table>
<thead>
<tr>
<th>Group Id</th>
<th>Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Variability OR variance OR variant</td>
</tr>
<tr>
<td>2</td>
<td>Hierarchical OR hierarchy OR &quot;separation of concerns&quot;</td>
</tr>
<tr>
<td>3</td>
<td>Model OR modelling OR modeling OR management</td>
</tr>
<tr>
<td>4</td>
<td>Guide OR methodology</td>
</tr>
<tr>
<td>5</td>
<td>Feature model OR Feature modeling</td>
</tr>
<tr>
<td>6</td>
<td>Product Line OR Production line</td>
</tr>
<tr>
<td>7</td>
<td>Automotive OR embedded OR large scale</td>
</tr>
</tbody>
</table>

Table 26 Keywords
6.1.2 Selection of Papers

To decide if a paper is relevant for this study first the title was evaluated then keywords, abstract and conclusions. If the ideas presented matched our criteria, then the article was considered for later processing. More thorough processing was done in the following way: fast look through the main part of the paper, concentrate on relevant ideas, make notes.

The paper was considered relevant if data can be used to answer at least one of the questions (combined by OR):

- What hierarchical feature modeling approaches exist in literature?
- Is the approach applicable for large scale product line?
- Does the paper provide methodology guidelines?
- Does the paper describe how feature modeling was done in industrial setting?
- Does the paper describe feature modeling (variability management) approaches in automotive industry setting?
- Does the paper provide solution to one of the problems of the case organization?
- Does the approach have tool support?
- Does the paper describe strategies to structuring of domain?

Articles that provided approaches to feature modeling, but did not provide technological support or the idea could not be compatible with GEARS tool were excluded.

6.1.3 Data Collection

To manage data Zotero Citation and Bibliography Processing tool (http://zotero.org) was used. This information was collected for each item:

- Full reference
- Category (feature modeling techniques, tool support, case study)
- Summary of main results
- Whether the study is applied to automotive industry
- Notes on if the paper results can be used by case organization. Precisely, issues of
  - Correctness
  - Completeness
Understandability
Maintainability
Scalability
Traceability
Redundancy
Modularity
Tool support

6.2 Results

During the literature review data retrieved from articles fall under one of the following categories: problems connected to variability management and feature modeling, product line structuring approaches, feature modeling notations, tactics used to organize the modeling space, guidelines on feature modeling and literature reviews in the field. The overview of the main contributions is presented in the next sections.

6.2.1 Literature Reviews

Systematic literature reviews retrieved from search [20,21,43] were a good starting source of information. These papers gave an overview of most well known variability management and feature modeling techniques (FODA, FORM, Kobra, COVAMOF and others).

Chen and Ali Babar conducted a systematic literature review of the variability management approaches, their effectiveness and level of evaluation [21]. They concluded that most of the methods were not scientifically evaluated enough and the quality of the evidence is low [21]. Among identified 91 approaches only 26 were evaluated in industrial setting. Moreover, Chen and Ali Babar claimed that it was not clear from the report if these methods were adopted in industry [21]. Other important finding is that almost 96% of analyzed approaches were evaluated by the developers of these approaches, which implies certain degree of bias [21]. Their research showed the need of further investigation in this area and the need of evaluation of the proposed methods in industrial practice.

Sinnema and Deelstra [67] classified and compared six variability modeling techniques (VSL, ConIPF, COVAMOF, CBFM, Koalish and Pure::Variants) and illustrated their applicability on an illustrative example. However, authors also agree on the fact that scalability of the techniques is not considered and the process is not described [67].

Authors of [43] conclude that 80% of studies that report practical experience with applying the method lack empirical evidence. Thus, it is hard to objectively compare and evaluate approaches [43]. Also, one of the interesting conclusions is that new methods and approaches are being created from the scratch and very few methods are being used and evolutionized. All papers [43],[21] and [67] state the problem of the validity of evaluation results.

As in this thesis attention is paid to the scalable approaches that can help with managing the product line complexity, the look was taken into “multiple product lines” [39]. This literature review highlights techniques used in organizing modeling space as a set of product lines. These product lines can be assembled together in a number of ways – hierarchical PL [37][73], model fragments [29], multilevel feature trees [61], context variability model [38]. This literature review [39] shows that the concept of multiple product lines to manage highly complex domains is getting more and more popular. Nevertheless, a new set of problems appears and supporting technologies should be developed.
6.2.2 Feature Modeling Notations

One of the points to discover through literature review was notations that are used to model features and comparison between them. The idea behind gathering papers, that not only explain the notation, but also critically evaluate different approaches, is to find out the concepts that are needed to technically realize approach. At the end, during synthesizes of the approach for the case organization, we need to state what concepts should be supported to successfully create a feature model.

The list of requirements to FM notation proposed by [32] are

1. Readability, as ability to graphically visualize the information. It is characterized as clearness and minimality of the model.
2. The notation should be simple and expressive
3. The notation should be able to express different types of variability
4. It should allow specifying properties of variation points
5. It should allow specifying dependencies between variable parts of Product line
6. It should allow easy changes to the model, adding new requirements
7. Support the traceability of models in time
8. The notation should be scalable
9. The notation should be supported by a CASE tool
10. The notation should fit into the whole development cycle
11. The notation should be standardizable.

This list of criteria is very similar to the requirements to the approach that were gathered with the practitioners in this work.

Feature modeling techniques that are based on the notion of feature are FODA (Feature-Oriented Domain Analysis) [41], FORM (Feature-Oriented Reuse Method) [42], FeatuRSEB (Feature/Reuse-driven Software Engineering Business) [35], GP (Generative Programming) [25], FORE (Family Oriented Requirements Engineering) [69]. However, they all share the common concepts, as modeling mandatory and alternative features and dependencies between them.

FODA [41] was introduced in 1990 and thus most of other FM techniques are built upon it. The initial proposal of Kang et al. was part of the Feature Oriented Domain Analysis (FODA) methodology. Its main purpose was to capture commonalities and variability at requirements level. This notation has the advantage of being clear and easy to understand. Unfortunately, it lacks expressiveness to model relations between variants or to explicitly represent variation points. Consequently, several extensions were added to Kang et al.’s original notation.

Griss et al. propose FeatuRSEB [35], a combination of FODA and the Reuse-Driven Software Engineering Business (RSEB) method. The novelties proposed are: introduction of UML-like notational constructs for creating FDs, explicit representation of variation points and variants, explicit graphical representation for feature constraints and dependencies.

Riebisch [62] states that explicit representation of cardinalities in FD is very important and so he adds UML multiplicities. Group cardinalities are introduced and denote the minimum and maximum number of features that can be selected from a feature group [62]. There are two other changes: a feature is allowed to have multiple parents; edges are made optional or mandatory, not the features themselves.

Czarnecki et al. studied and adapted feature models in the context of Generative Programming [25]. The difference with FODA is the OR feature decomposition and
definition of a graphical representation of features dependencies. Also, the notation was extended with the concepts of staged configuration (used for product derivation) and group and feature cardinalities.

The studies show [9,32] that feature modeling notations describe types of features, dependency types, cardinalities and properties of the features. However, the approach cannot be built only on the chosen notation. There should be some guidelines on structuring the modeling space and tools that should be used.

6.2.3 Guidelines

As stated in the literature reviews and proved by industrial interview survey methodological guidelines are not provided consistently by authors of the approach. Some of the advice can be extracted from different papers [29,36,50,68]. This section will give short overview of existing methodologies.

6.2.3.1 Feature Identification

Features can be identified according to feature categories framework [50]. With respect to FORM features can be classified into four categories: capabilities, domain technologies, implementation techniques and operating environments. Subcategories are shown on Figure 14. However, it is not exactly explicit what feature belongs to which category and how to organize these categories. The example feature model using FORM feature groups is shown on Figure 15. Private Branch Exchange (PBX) is a telephone system that switches calls between users within a small area, such as a hotel, an office building, or a hospital [50].

Another guidelines from [50] are:

- “Try to first find differences among products envisioned for a product line and then, with this understanding, identify commonalities”. The practice shows that usually there are more commonalities than variabilities in the system. That is why it is easier to identify what is different among product variants.
- “Do not identify all implementation details that do not distinguish between products in a domain”. Feature model is not a requirements model and should concentrate on the differences, but not on all the details of the system.

![Figure 14 Feature Categories][50]
Yet another method to identify features is a use case based approach described in [74]. This approach consists of three main steps:

1. Construct a set of feature models for individual applications (called application feature models, AFMs) from use cases of these applications, with the support of a set of discovery rules;
2. Adjust the AFMs with the help of a set of adjusting rules and conflicts checking rules;
3. Merge the set of adjusted AFMs to form a feature model for the domain (called a domain feature model, DFM).

The method presented is semi automatic. The idea to review use cases of the system to identify differences in product variants can be used as a guideline as well.

6.2.3.2 Feature Organization

As previously mentioned, important problem is how to organize features in the feature model. Lee et al [50] advises to organize features in order to show commonalities and differences, but not to represent functional dependencies. Feature diagram should easily show how feature can differ among products. Another valuable notice from [50] is “If many upper-level features are associated with a lower-level feature through the implemented-by relationship, reduce complexity of the feature model by associating the lower-level feature with the nearest common parent of its related upper-level features.” Feature model should be refined to reflect architectural boundaries [50]. By doing so it is easier to achieve traceability among different artifacts.

Features can be also structured according to different criteria [29,36]:

1. Mirroring the solution space. This approach is good especially when creating a feature model for already existing software. The technical solution structure can
be reflected by feature model. For instance, separate feature models can be created for different subsystems of a product line. “Similarly, the package structure of a software system or an existing architecture description can serve as an initial structure. The number of different models should be kept small to avoid negative effects on maintainability and consistency” [29,36].

2. Decomposing into multiple product lines. Complex software intensive systems usually make use of “system of systems” structure, creating product line of each subsystem of the product line. For example, there may be separate product lines for different target customers, e.g., mobile phone product lines for senior citizens, teenagers, and business people. Multiple product lines can also be structured differently and using different technical solutions [39]. Decomposition into multi product lines will be discussed in Section 6.2.4.

3. Structuring by asset type. This type of structure deals with creation of variability models for each of the asset type in the system development. Possible assets are: requirements specification, system architecture and design, test specifications, documentation and others. For example, feature models can be created separately for requirements (requirements variability model), architecture (architecture variability model) and documentation (documentation variability model).

4. Following the organizational structure. This approach means to structure product line models based on the team structure of the organization. Thus, it may seem as a good idea, it can lead to redundancy in feature models. Flores et al [33] proposes to structure mega scale automotive product line according to organizational structure.

5. Considering cross-cutting concerns. This approach is based on concepts from aspect-oriented development. For instance, aspect-oriented product line modeling can be used to model both problem and solution space variability. With this decomposition a model of core features that all products have in common is created and also aspect variability models are defined for product specific features [29,36]. Complex aspect dependencies can lead to difficulties with regard to managing their interaction [29,36].

6. Focusing on market needs. Within this approach marketing and business needs drives the structuring of the modeling space. This approach makes easier the communication with the customer and among business units. However, this leads to models that are completely unrelated to technical solution.

Of course, the combination of the approaches are also possible [29,36]. The approaches stated above can be extended in the context of different organizations. These strategies can be referred to structuring in the large. When thinking about organization of features (or functions) inside one feature model (structuring in the small), the number of tactics can also be thought of. Nevertheless, this is not addressed properly in the existing literature.

Some advice on structuring in the small can be found here [50,52]. The papers discuss ambiguities and unnecessary complexities during modeling.

### 6.2.4 Approaches for Structuring Large Scale Product Lines

As mentioned before vehicle product lines can be characterized as large scale or mega scale product lines [33,39]. To reduce their complexity these systems can be modeled as product line of product lines (hierarchical PL) or also referred to as multi product lines. MPL can be defined as “a set of several self-contained but still interdependent product lines that together represent a large-scale or ultra-large-scale system” [39]. Following subsections will give short overview of the most promising approaches. The main idea behind all the papers is reduction of complexity of a big feature model by means of decomposition. The
approaches propose different methods to achieve the goal. The details of each method could be found in the cited papers.

6.2.4.1 Multi Criteria PL

Product lines in automotive domain can be considered as multi criteria product lines [17]. In this kind of systems products can be formed according to different criteria: geographical location, cost, vehicle line and so on. Many organizations setup product lines for each of the market criteria [17]. However, this causes even more complexity due to redundancy of features. This leads to inconsistencies and hard maintainability.

Buehne et al in [17] presents an approach that allows to model variability in a single model for a multi criteria product line. The approach is based on the common feature model for all vehicle line and a set of assignments of features to different vehicle lines. Common feature model consists of all features of all vehicle lines and is based on FODA notation. Figure 16 shows an example feature model for a car part “Wiper”. The relation between feature model and market criteria (vehicle line in the given example) is call assignment dependency. This dependency is needed to define views on the common feature model and determination of feature availability in the specific vehicle line. The notation of assignment dependency is shown on Figure 17 together with usage examples.

![Figure 16 Common feature model - for all vehicle lines][17]

---

[17]: https://www.example.com
The proposed approach has its own benefits as clear definition of what is common to all vehicles, what is variable and mandatory to a specific vehicle, views can be defined, better maintenance. However, so far the visualization mechanism cannot be suitable for a large scale product lines. The other important drawback is absence of the supporting tool, which is an essential criterion for an industry adoption of the approach.

6.2.4.2 Multi-Level Feature Trees

As another solution to managing high complex product lines is the approach proposed by Reiser and Weber in [11,61] in the context of research project in DaimlerChrysler. The main goal is to hierarchically organize product sublines (vehicle groups, brands, lines, models). The concept is based on the notion of reference feature model [61]. Reference feature model is a template or guideline that defines default features, their properties and deviations [61]. Reference feature model is basically a usual feature model with only one difference. There is a feature model that references it – referencing feature model. The example provided in the paper is shown in Figure 18, where “Series Cluster” is a reference feature model and models “Series A”, “Series B” are referencing models. The paper also defines variations that can occur in referencing models and usage scenarios.

The drawback is again the lack of tool support.
6.2.4.3 Decomposing Feature Model into Parts

Obviously, maintaining one feature model for such a domain like automotive is not feasible. There are attempts considering for decomposition of a single feature model into parts [2,3,6,12,29]. Papers present formal definition of the proposed approaches based on slicing operator [2,3], model fragments [29], set representation of product lines [73], Distributed Description Logics [6]. The main goal of each approach is to find meaningful decomposition of features into feature model modules, the mechanism to establish traceability between modules, merge and configuration operations.

Model fragments [29,39] are partial models with defined dependencies to other models and should be merged together to form a complete model of the product line. The approach proposes also a tool support for semi automatic merge of model fragments.

Thompson and Heimdahl [73] structure product line as a set, “where the boundaries of the set are determined by the commonalities, and the individual members of the set are distinguished by the values of their variabilities” (Figure 19).

The idea of decomposition of feature model into parts is also described by [2,3] and [6]. [2,3] present a slicing operator, which works similar to the concept of program slicing. First, the slicing criterion is defined (set of features that are relevant for SPL practitioner) and then a new feature model is computed [2,3].
Together with benefits come problems. These include lack of tool support, no methodology on how actually to modularize models, redundancy and overlap support.

### 6.2.4.4 Models with Context Variability

Context, as a set of market criteria, is important for each product line. In [38] the approach proposes to distinguish between context variability model and the traditional feature model. The combination of two allows realizing complex multi criteria product lines. Context variability model describe variability of the environment, thus context. It can be defined in FODA notation (or any other feature diagram notation), but does not define features in the traditional sense [38]. The dependencies between features and context are modeled as dependencies between feature model and context variability model. The example of the approach is shown in Figure 20 presenting context variability model and a feature model of the infotainment domain of the vehicle [38]. In this way feature models are reduced in size and complexity.

Figure 19 Hierarchical decomposition and subset structure [73]

![Hierarchical decomposition and subset structure](image)

Figure 20 Multiple product line feature diagram of the infotainment domain [38]

![Multiple product line feature diagram](image)
6.2.4.5 View Based Feature Modeling

To manage the complexity of feature model it is also possible to define views on the FM according to the stakeholder needs or market criteria [5,17,34,64]. Views can be helpful in improving understandability of the FM as they reduce the size of FM and tend to show only relevant features, variants and dependencies. Mechanism of view can be imagined as a filter defined upon a complete feature model. [5] separates variation points to local and essential. Thus, essential variation points are common for all stakeholders and are defined mostly by business needs. Local variations are defined for different stakeholder groups (developers, architects, testers and so on). Essential variations should be managed consistently and communicated to all the stakeholder groups.

The paper [34] deals with managing variability across different models (use case model, collaboration model, feature model, state chart model).

Already described approach to modeling multi criteria product lines [17] aims also to define views on the common feature model by defining assignment relationship between feature model and market criteria.

6.2.5 Problems and Evolution of Feature Models

Authors of [9] analyze properties of the universal approach to variability management. As feature modeling is one of the popular techniques this paper states problems of feature modeling with respect to traceability, scalability, consistency and visualization [9]. The article is skeptical that feature model by itself is enough to fulfill these requirements to variability management approach. The same problem is described in [16]. They both agree that feature model is helpful in describing and communicating variability, however, other techniques are needed to ensure consistent variability management in all phases of software development lifecycle.

Yet another problem with feature modeling in particular is differences in representation of the same semantics even within one notation [50,52]. This kind of ambiguity happens because of different understanding of variations by feature owners and absence of modeling guidelines. Nevertheless, [50,52] present a set of ambiguities that can happen during modeling and advice on reducing complexity of feature models. Some of these tips will be used in the guideline proposed to the case organization.

6.3 Discussion and Summary

To move forward with the construction (or selection) of the feature modeling approach in the case organization it is therefore necessary to have information about current directions of feature modeling. This was achieved by reviewing literature in the field. It was decided to make search through electronic databases and conference proceedings in software product lines and variability management. Special interest was for papers in the automotive domain or other embedded complex domain. “Snowballing” technique was applied to make sure that relevant papers are not missed. The selected approach proved itself well. It allowed concentrating on the presented ideas to create an overview of the feature modeling strategies. Though, state of the art was described. The mind map showing grouping of results are presented on Figure 21. The mind map shows three directions to approach feature modeling and structuring. First of all there are structures. As we see there are three possibilities to structure a feature model:

- Single feature model for the whole domain (Monolithic FM)
- Division of feature model into modules (Modular FM)
- Multiple Product Lines
Another direction of discussion is tactics, or criteria, that can be applied to structuring a feature model. Tactics can be applied to any of the three proposed structures. To distinguish between these two directions it is proposed to use word “tactics” for criteria or principle of structuring a FM, and “structures” to refer to special organization of feature model (monolithic, modular, MPL).

In this literature overview we also gave a short introduction to feature modeling notation, but did not concentrate on description of syntaxes.

The results presented here show that an effort to create a universal feature modeling approach is made. Practitioners also see the advantages of FM, however problems arise in the complex domains. The lack of experience in modeling and unavailability of industry approved tools are the reasons that prevent successful application of FM in the industrial setting. As was assumed in the beginning of the thesis, domain specific modeling guidelines for industry are not available. There are very few papers that try to summarize their experience with modeling. However, we see that none of the techniques are a standard industrial practice.

This overview shows the techniques that can be applied to organize feature model, the choices that can be made by industry practitioners.

Methods obtained during the literature review should be put in the context of automotive company. In the next chapter the suitability of the proposed approaches in the setting of the case organization is discussed and also domain specific decomposition tactics are added.

![Figure 21 Literature Review Mind Map](image)

The mapping between requirements identified in Section 4 and approaches to structuring feature model is given in Appendix 2. The tool support is a weak point of structuring approaches. Only some of them can be modeled within GEARS and this will be discussed in the next chapter. It can be also noted that not a single approach satisfies all the requirements. That is why the adjustments or combination of the approaches should be done.
7  INDUSTRY EVALUATION OF FEATURE MODELING APPROACH

As seen Feature Models can be structured in different ways, whereas current literature approaches emphasize on notation, configuration and tool support. This chapter will present a number of ways to structure a product line and discuss problems and inconsistencies that can happen during modeling. The selection of structuring approaches was inspired by state of the art, but also the domain context. Feature modeling patterns were collected from literature review, from interviews with stakeholders and brainstorming. According to the industrial goals, stated in the Section 5.3, the possible solutions will be given. Models were discussed in the technology independent level to assure that the reuse is possible. Of course, some aspects depend a lot on the tool chain used, but the structuring principles should remain the same from tool to tool. It is expected that the tool selected by the industry (GEARS from BigLever) will have some deficiencies and would not provide comprehensive mechanisms for model implementation. However, the approach should be adjusted to the selected technology.

7.1  Research Method

The goal of this chapter is to put approaches selected during literature study into context of automotive industry, and case organization in particular. It is aimed to see if and how the proposed solutions fit into the case organization, if they can be combined in order to satisfy the stakeholder goals and what benefits can they bring. To answer this question we could design an experiment to model using different techniques and tools. However, the problem was an absence of the initial subsystem and requirements to it. Also, not all of the selected tactics and structures could be applied to the same example. Some of them are applicable to the whole product line, some only to the subsystem (or a single feature model). Another factor was a time frame. Formal experiment, together with quantitative data collection and involvement of practitioners, would require more time than was available. However, the recommendations on gathering metrics can be used in experiment design in the future. In this thesis we carried out a case study, during which we gathered possible solutions, structures and tactics and discussed them in the context of case organization.

Data collection method used was regular workshops with practitioner from the case organization, who is responsible for introduction of ‘New Generation Tool Chain’. During these workshops various decomposition techniques were brainstormed and put them in the automotive context. Later on benefits and tradeoffs of the techniques were defined that allowed to decide on the final approach that should be followed in the future in organizations. The decomposition approaches deal with decomposing the whole system. The approaches were tried to apply on the higher level of abstraction by researches and discussed with practitioner during meetings. Based on these workshops a guideline were also created, which should be further evaluated by practitioners.

The research process consisted of the following steps:

1. Discuss and do high level tool independent modeling with practitioner using different structuring approaches (monolithic model, modular FM, MPL)
2. Gather benefits and drawbacks of the structuring approaches. Based on this information draw conclusions about best fit decomposition for case organization

To put tactics into the context of automotive engineering we followed this process:

1. Brainstorm additional decomposition tactics valid for the case organization and automotive systems in general
2. Discuss with the practitioner possible implementation of the tactic
3. The researcher tried to model using the selected tactic
4. The results of modeling were discussed with practitioner and documented by researcher
5. The practitioner reviewed the summary of benefits and drawbacks of the tactic with respect to automotive industry
6. The previous steps were repeated for all tactics
7. The tactics that suit case organization and can be combined together were selected to be included in the final approach.

The combined approach was applied by researcher and reviewed by practitioner to the subsystem of case organization “Parking Assist (PA)”. The feature model for PA was created and also the infrastructure of product line to illustrate the approach application. The modeling was done inside GEARS tool.

The main threat is external validity, as the results obtained in this study might not be applicable in the context of other organizations or domain. However, the methods followed can be reused.

The threat to construct validity is that the evaluation and selection of the approach is done only by one person. The researcher could be biased or have the wrong view on the industrial setting. To mitigate the threat the discussion with the practitioner was done before and after applying the tactic or structuring technique. The results could also be biased by preselected tool.

The reliability threat deals with results repeatability. To make sure that the same conclusions can be drawn by other researchers given the same setting the reasoning behind selection of tactics, explanations of benefits and drawbacks were documented.

### 7.2 Approaches to Structuring Feature Models

Starting to build a feature model of the Vehicle first of all it is needed to decide on the strategy of refinement and structure of the whole product line. As noted from the literature review models can be decomposed and arranged in the three major ways:

- Monolithic model
- Modular model
- Multiple product lines

Next sections describe benefits and drawbacks of each option in the context of case organization.

#### 7.2.1 Monolithic Feature Model

Before describing possible hierarchical structures monolithic feature model will be discussed. By monolithic feature model a single feature model of the whole system can be understood. There are no other feature models of subsystems and components. Although having one model may seem easy and comprehensive there are a lot of problems connected to it.

Considered benefits are:

1. All the dependencies and constraints between features at different levels can be modeled.
2. No restrictions on feature visibility. In case of single feature model features and subfeatures have global visibility, not limited to subsystem or component (as in case of decomposed FM).

3. Features at different abstraction levels are easily traceable.

Monolithic feature model can be good for small applications, where the number of features and their variants does not get in the way of understanding it and maintaining it properly.

The following problems have been identified of having a single FM in the case organization:

1. The biggest threat to understandability and maintainability is the huge number of features and variants in the automotive domain. The approximate size of automotive system was discussed in the background chapter. No human being can process this amount of information at once. The model will not fit in the screen, so that it will be hard to analyze dependencies in such a model.

2. Features in the model cannot be reused. If some feature is required multiple times and the feature model is a tree the feature should be duplicated. This creates unnecessary complexity, the model is hard to maintain and understand. Even if FM can be a graph, arrows from different parts of the model can be hard to follow.

3. No mechanism for separation of concerns and work separation.

4. Global impact of all features [47]. This means that a feature can have impact on any other feature, and vice versa. This dependency is especially bad for managing evolution. Features should be modeled in their own scope.

5. Valid configuration is defined by making decisions on all the variants. This means that the person responsible for definition of configurations on FM should be aware of all the variation points and all the details on all the abstraction levels. Again, because of domain complexity that is not feasible.

Even if monolithic model is structured in the best possible way the problems will overweight benefits. Some techniques that can be helpful in managing monolithic model:

1. Defining mechanism of views. Views can help in selecting of a subset of features relevant only for a given point of view. This can improve understandability of a monolithic model as it is obviously easier to process only relevant information.

2. Locating related features (by any kind of relation) close to each other. However, positioning can be a problem also, as a magnitude of Vehicle Feature model is not 10 or 20 features, but thousands.

Schematically monolithic model is presented in Figure 22. The system in case organization is defined on the three abstraction levels (L1, L2, L3). Feature model is the same for all the levels and all subsystems. Traceability links to requirement documents can be defined.
7.2.2 Modular Feature Models

By modular feature models this thesis understands feature models that are divided into parts, which are hierarchically arranged to form a complete FM. Forming a hierarchy can make the life of involved stakeholders easier. The points to discuss are level of granularity, structuring principles in large, structuring principles in small.

One can imagine modular FM as a set of smaller feature models, that can be combined together and form a FM of the end product. Theoretical aspects of modular decomposition of feature model are shortly presented in Section 6.2.4.3. Figure 23 presents schematic decomposition into modular feature models. Feature models are created on different levels of abstraction (and possible different features). Traceability links go from feature model to requirements artifacts, design artifacts and through the whole V model to verification and validation artifacts. The traceability should be assured automatically by tools.

As we see it, the benefits of decomposing feature model compared to monolithic model are:

1. Easier to understand. The less number of element in feature model the easier it is to understand it and dependencies between elements. Also, important is that users do not have to scroll through the whole single model to find the relevant features or parts.
2. Easier to maintain. The impact of change is limited to one smaller model.
3. Easier to create, when the decomposition principle is clearly defined. Models could be created by different people, precisely by feature owners or specialists in the defined area.

4. Enhanced reusability. Features can be accessed by reference and not by copying. This quality contributes also to better consistency and maintainability.

5. Reduced complexity. Cognitive complexity is reduced due to better separation of concerns.

In the case organization the decomposition boundary can be a feature (in the broad sense). For example, a feature can be Parking Assistance, or Climate Control.

However, managing modular feature models can bring up some problems and challenges as well. Mostly there should be tools that can combine parts together and maintain dependencies between different models. The next point summarizes problems and challenges connected to modular FM in automotive domain:

1. Big amount of modules that lack overall structure. When decomposing according to customer visible features, the overall structure could be lost. There is a need to group decomposed feature models with respect to some criteria (for example, region or package structure).

2. Maintaining traceability between models.

3. Configuration of models. The benefit of modular feature model is their easier and faster configuration. However, a new technology is needed to define and combine configurations to form a valid product. For example, in the case organization the model for park assist, climate control, sensor fusion and so on can be defined. Afterward, valid configurations of features should form a valid configuration of the product (vehicle system).

4. Dependencies between features of different feature models. The usual mechanisms can be used to define dependencies (requires, excludes, various assertions) inside one feature model. However, it can occur that features from different feature models can be dependent. It should be technologically possible to define such relationships.

As it can be seen modularity brings a lot of improvement to feature models in automotive domain. However, we need an extension in FM notations and tools to allow combination of modular FM into a valid product.

### 7.2.3 Multiple Product Lines

As vehicle consists of many different parts that can form a product line by itself it is considered a good idea to start multiple product lines that are hierarchically assembled to form a valid product. However, these different product lines interact with each other, require similar features and depend on the same top level choices. The tradeoff between modularity, understandability and completeness should be understood.

Top level product line represents the end product that is namely a car (Figure 24). Each car can be refined according to several criteria, contain different parts and be issued for different markets. The best decomposition technique depends on the needs of stakeholders and the context of the company.

Establishing product line of product lines brings the following benefits to the case organization:

1. Improved separation of concerns. Each feature owner, or supplier, can work with their own product line. There work does not interfere with each other.

2. Improved overall structure, compared to modular feature models.
3. All the benefits of modular FM, as basically single feature model are also decomposed following this approach.

4. Multiple product lines also favor a step by step adoption of feature modeling. In such a domain as automotive it is nearly impossible to switch the whole development to the new approach immediately. That’s why, with multi product line approach the switch can be done piece by piece.

One of the problems of this approach is cross system functions and managing of dependencies across product lines. This should be satisfied with the selected tool.

![Product Line of Product Lines](image)

**Figure 24 Product Line of Product Lines**

### 7.2.4 Conclusion

Based on the discussed benefits and drawbacks of the three decomposition techniques, the resulting approach was selected by practitioner during the workshop. It was discussed that for such a complex systems as Automotive structuring product line space in multiple product lines is the better option. This decomposition is natural due to the domain specifics. However, decomposition can be done further into product line of customer features. It was proposed by the researcher and approved by the company’s practitioner that further decomposition can be beneficial. As the development is done on three abstraction levels, each level can be modeled as a feature model inside one product line. Combining modular approach and multiple product lines can be beneficial for better structure and traceability. The combination of two approaches is presented in Figure 25, where each product line can consist of several feature models. Solutions to the arising problems can be found by establishing processes, tools and guidelines.
Previous section discussed the choice of the structure in the large. It is showed that for the case organization beneficial is a combination on modular feature models and multiple product lines. However, decomposition can be done according to various criteria. It is referred to them as tactics. In Section 6.3 in Figure 21 the tactics retrieved from literature study are summarized. Here they are discussed form the point of view of automotive industry and case organization. Some other decomposition criteria were added to the list.

### 7.3 Tactics

The starting point in decomposing a PL is the existing information and approaches of the company. It is considered a good practice to introduce new process descriptively. When introducing an already familiar to the personal structure it would be easier and faster for them to accept it. In the case organization the basics for all development is functional architecture that divides whole system into domains, subsystems and features (Figure 26). The product line can be decomposed into multiple product lines with respect to domains (active safety domain, infotainment domain, chassis domain and others). Domains can be further decomposed into subsystems and system features (parking assist, cruise control, climate control and others). This example decomposition is shown in Figure 27. For illustration purpose only 3 domains were decomposed. This level of granularity is good enough, as the models are not that huge and correspond only to one feature.

The benefits of decomposing PL according to functional architecture are:

1. Functional architecture is the basis of all development processes in the case organization.
2. People minds are set to view the system as decomposition to domains, functions and subsystems.
3. Allows separation of concerns, product lines can be maintained by people responsible for a specific feature.
4. Allows work separation. Work can be done in parallel and integrated later on.
5. Function architecture also reflects solution space structure.

The challenges connected to this decomposition are:

1. Cross cutting concerns and non functional variability. The open question is how to cope with cross cutting concerns (such as brand, country or marketing unit) and non functional variability. The solution proposed for case organization is to define the ‘main’ domain of the cross cutting function and then reference this functionality from other domains.

2. Approach to minimize redundancy is needed. Some feature and functions can be used in more than one product line. The requirement is to minimize redundancy, which means that features should not be duplicated, but accessed by reference.

3. Related problem to 2nd point is coping with Vehicle Option Codes that are global for the whole vehicle and used to define configurations across product lines.

4. Traceability between different levels. The mechanism to define dependencies between different product lines and feature models of different abstraction layers should be established.

![Functional decomposition diagram](image)

Figure 26 Functional decomposition
7.3.2 By Organizational structure

Structuring a product line according to organizational structure makes sense, as it allows separation of concerns for different stakeholders. In the recent paper [33] it is argued that organizational structure is the one that fits the case organization. Feature owners do not need to see all the details about other features and their variations. So, the feature model can be modularized according to organizational structure. In the case organization the separation is between Executing Organization and Reuse Organization (Figure 28). Executing Organization consists of departments that are responsible for various Vehicle Programs that in their case are decomposed to Domain teams, subsystem engineering team and feature owner teams. As we see, after case organization is aligned to Functional Architecture. The Reuse Organization departments are responsible for reusing of the work done by the executing organization. Reuse Organization consists of Bill Of Materials Leads (BOM), BOM Family Owners and Design and Release Engineers. So, basically executing organization is responsible for defining the feature models and producing a product according to it, whereas Reuse organization uses the feature models to configure the end product.

As already mentioned organizational structure corresponds to functional architecture. So the result of the previous section should not be discarded. However, the reorganization happens overtime. The effort of reorganizing feature models is high, as not only the hierarchy should be changed, but also dependencies and assertions.

So, to conclude the benefits of organizational structure are:

- Separation of stakeholder concerns
• Separation to domain engineering (Executing organization) and application engineering (Reuse organization)

• Aligned to functional architecture in our case, so all the benefits (and drawbacks) of the previous approach

The drawbacks are:

• Frequent company reorganization.

![Organization](image)

**Figure 28 Organizational Structure in the case organization**

### 7.3.3 Mirroring the Solution Space Structure

In Section 6.2.3.2 it is shortly described organization of features according to the solution space. This means that different feature models should be established for subsystems, or package structures.

The solution space of the studied organization consists of subsystems and components. Each component can require ECUs, sensors, cameras and other type of hardware, as well as software that is deployed to an ECU. So, features are combined into packages and deployed to ECUs.

The benefits of this tactic are

• The clear overview of what features are packaged, deployed and tested together, because these features are grouped together in the feature model.

• Traceability between feature models and components is easier to manage, as FM reflects the component structure. However, in functional decomposition domains usually form a package. The consequence is that feature model of the package is too big to maintain.
During the discussion with the company’s practitioner this decomposition was not considered applicable to the case organization for several reasons:

- Problem and solution space is not in one to one dependency.
- The feature models, at least in the beginning, are aimed for requirements artifacts. They are not aligned to technical solution.
- It is not excluded that packaging of components can be changed. Components can be regrouped or form a new package. These changes will require changes in the feature models and dependencies between them.
- Another difficulty is features that are distributed over several components. The problem with redundancies and inconsistency then arise. The amount of rework needed for these changes is expected to be high.

That is why decomposition according to the solution space structure was ruled out.

7.3.4 By Asset Type in the Domain

Variability information can be captured in different feature models according to the asset type which each describes. This concept is similar to the concept of separation of concerns, where views are defined to reflect special quality of the system. Thus, separate feature models can be created for requirements specification, architecture design and so on.

For example, architecture variability model can contain architectural choices regarding type of ECUs (4, 6, or 8 channel ECU for Park Assist) or deployment to ECUs (several choices on which ECU feature should be deployed). Also, these choices depend on the car model.

The benefits of the tactic are:

- This approach brings clear separation of concerns by representing only relevant variations of the features.
- That also at the same time tends to reduce complexity by hiding unnecessary information.
- Moreover, models can then be structured differently, representing the asset structure for instance.

The biggest challenge is the duplication of information. Even though, different assets concentrate on different aspects of features they still have common concerns as well. These common parts should not be duplicated. As discussed before duplication leads to inconsistencies and maintainability problems. However, the technical solution can be found. The GEARS tool provides concepts that can be used to overcome this challenge. This will be discussed in the GEARS implementation chapter.

7.3.5 Focus on the Market Needs

Decomposition of product line according to vehicle brand, region, car line, car class and other market criteria is represented in Figure 29.

Product lines that take into consideration marketing decisions, make it easier to communicate variability to customers and business departments. They also would need a common family model, that would consist of features (common and variable) which are there in every car line or car brand.

With this kind of structure features from different series, classes, brands will need to be duplicated several times. No duplication of information is one of the industrial requirements. That is why this structure as it is, does not satisfy the requirements. However, research
approaches such as family modeling, reference feature modeling and others will help to realise it. Nevertheless, above mentioned approaches are not well tool supported.

Figure 29 Decomposition according to market criteria

7.3.6 By Physical Structure

In this section we will look into structuring product lines according to physical architecture of the vehicle. Figure 30 shows the structure of the Park Assist Control module. The vehicle can be divided into parts as body, wheels, breaks and many other components. As a benefit of this decomposition is

- Clear vision on deployment information, dependencies between functions and hardware information.

On the other hand this type of structure is complicated to maintain in the case of change. Once the feature is deployed to the other ECU or component the feature model should be changed. There is no independence between components and features. Though, case organization tries to switch to AUTOSAR [19] architecture to make software independent of deployment. That is why this tactic will not be adopted in the case organization.

Figure 30 Physical structure of the Park Assist Control Module
7.3.7 By Stakeholder Concerns or View Based Decomposition

Defining views on feature model based on stakeholder concerns

- Reduces cognitive complexity by providing only specific relevant information to the user. It would be nice to have a common feature model and a filter mechanism that would work over it. This would keep the models consistent, but also hide unnecessary information.

- As a result is the benefit of better understandability, maintainability and consistency, as well as separation of concerns.

As discussed in Section 6.2.4.5 variations can be essential, those relevant for all the stakeholders, and local, defined only for a specific user group. For the case organization essential variations are Vehicle Option Codes and some of the architecture variations.

Nevertheless, the biggest challenge is a tool support. Unfortunately, tool selected by the case organization does not support views (as a filter mechanism).

7.3.8 Evolution and Localization of Change

Another tactic that can be applied is modeling along the evolution and changes. If there are defined evolution scenarios that usually happen to the system, then these possible changes should be taken into account. By thinking in this way the system should be analyzed to discover patterns of changes, what modules are closely coupled with each other and what changes are possible in the future. For example, steering (chassis domain) and braking (chassis domain) depend on each other. Then by following this tactic the features of these modules should be modeled together. Another example is cluster (infotainment domain) and visible/audible feedback.

The benefit of this tactic is

- Trying to “foresee” the future. For instance, introduction of new technologies can be done easier if it is modeled already in the top of the hierarchy of product line. If the new technology is first introduced for one feature, but in some time it is expected to be a standard for other features as well, it is better to model it on the domain level to avoid changes to the hierarchy of feature models.

- Minimize impact of change and rework

The drawback is that in our particular case it is hard to initiate modeling for a large scale product line in this way.

- This is a new way of thinking and seeing the system. That leads to a double effort – adopting product line concept and changing the vision of the system.

- Another problem is that there is no evolution information to take into consideration, as these metrics were not needed before.

This approach can be used as a guideline to try to predict changes during initial modeling and try to organize features in the way that is easier to change.

7.3.9 Using Context Variability

In case organization one of the confusions is modeling of vehicle option codes. These are parameters that are global for the whole system. For example, transmission type of the vehicle or driving type (left hand or right hand). To model these features the concept of context variability can be applied. The context variability model forms a separate feature model of the top level product line and dependencies to other features models can be defined. On Figure 31 the structure of product lines with context variability is shown. Elements in blue can be modeled as product lines as well.
The improvements connected to this tactic are:

- This approach reduces redundancy, as vehicle option codes are defined in one place and are not copied.
- It contributes to consistency.
- This tactic is well applicable to decomposition according to functional architecture and can be used together.
- It also helps to conveniently define feature variability that depends on more than one criterion (e.g. region, car type, transmission type).

The challenges seen with this approach is possible big number of dependencies between context feature model and product lines of domains and features. The tool should support this kind of dependencies and structure. GEARS is compatible with modeling in this way.

![Figure 31 FM with Context Variability](image)

7.3.10 By Feature Category

Feature categories can be used to structure the feature model and not to forget important aspects of the feature. Feature category can be represented as a feature group in a feature model or only used as a guideline.

7.3.11 Conclusion

This section described tactics that were found during literature review and defined through workshops in the context of the case organization. Benefits and challenges were described and if possible the schematic examples were given. As seen, decomposition according to functional architecture fits the organization best. Nevertheless, other tactics can be applied together to solve some of the problems in the domain. No single strategy is a perfect solution; the approach should be synthesized with respect to problems, selected tools and goals.
Based on the discussed benefits it was seen suitability to combine following methods. First of all Context Variability model should be established to gather all the global parameters of the vehicle. Secondly, variability models that correspond to assets in the domain should be created. Moreover, feature categories should be used as a guideline on describing all the relevant aspects of the feature. The other tactics were discarded as in this specific context their challenges outweigh benefits.

Next section presents the GEARS concepts available for modeling, GEARS specific implementation of the above mentioned tactics and structures, and modeling results of Parking Assist feature.

### 7.4 Feature Modeling Approach with GEARS

#### 7.4.1 GEARS Modeling Concepts

Gears is a tool used to create a system and software production line capable of producing all of the products in a software product line portfolio [10]. The main concepts of Gears production line are [10,33,46]:

1. **Systems and Software Assets.** These are configurable artifacts – such as models, source code, requirements, and test cases with defined variation points.

2. **Feature declarations** are “parameters that express the diversity in the product line for a system or subsystem. Feature declarations typically express the customer visible diversity among the products in a product line. Feature declarations have types. When a feature is chosen for inclusion in a product, it must be given a value consistent with its type.” In the Table 27 types supported by GEARS are presented. Feature model can be edited and viewed in text, structure and graphical formats.

   ![Table 27 Feature Declaration Types in GEARS](image)

3. **Feature assertions** are constraints and dependencies among the feature declarations. Assertions can be of REQUIRES or EXCLUDES type. They express the constraint that a feature (or combination of features), if present, either requires or excludes the presence of another feature (or combination of features).

4. **Feature Profiles** define valid configurations of the product feature model in the product line. With the help of this concept it is possible to select features and define values of feature properties. The selection should not contradict to defined feature assertions.
5. The Product Configurator automatically assembles and configures the systems and software assets, guided by the product feature profiles, to produce each of the product instances in the portfolio\cite{10}.

Feature profiles are defined for each feature model in the assets. Feature profiles not only predefines choices to be made over a feature model, but also reduces the number of possible configurations from exponential to a linear list of valid feature profiles \cite{47}. This is a great benefit especially for testing assets, as only valid configurations can be validated. The Product Configurator actuates the asset using feature profile and defined variation points.

There are several concepts in GEARS to support composition and modularization of the system \cite{10,47}. These abstractions are:

1. **Mixins.** When feature declarations fall in more than one asset they should be modeled as mixins. Mixin allows to create feature declaration in one place, and not to duplicate it or copy in all the assets. Profiles also are defined for features declared in the mixin. The single location of profiles keeps the system consistent during configuration. Mixins can be also imported to another product line and reused in the other context.

2. **Matrices.** These are tables that help to build up the product by telling the configurator what feature profiles to use for each asset and each mixin in the production line. Each row of the table specifies one product. Each column corresponds to asset or a mixing. So that each cell defines the profile for assets and mixins in a product. Matrices also define value for imported product lines and mixins.

3. **Imported production lines** are used to create a hierarchy of nested production lines. The production line can be used on its own but also imported. It will then appear as a column in the product matrix.

Figure 32 shows a view on GEARS main concepts. A Production Line Base “Vehicle” is shown to consist of mixin “VehicleOptions” and imported production line “Active Safety”. Mixin “VehicleOptions” have two profiles defines: “EU” and “USA”. “ActiveSafety” production line defines two products (“Standard” and “Advance”). “Vehicle” defines a table of products “Car” that is built up of “VehicleOptions” and “ActiveSafety”. Using this information the end product can be actuated.

![Figure 32 GEARS main elements](image)
7.4.2 GEARS Specific Implementation of Selected Approach

To model the selected structure and tactics in the case organization it is necessary to describe the correspondence between GEARS concepts and selected tactics. This section will propose the way to implement selected approach with GEARS tool chain.

Table 28 shows this correspondence. Product line of product lines can be implemented as nested production lines within GEARS. So, the product lines of customer features are imported into product lines of domain, whereas product lines of domains are imported into top level product line. All product lines define a set of mixins that can be reused in assets and other product lines. A set of profiles are defined over feature model of mixin and valid products are configured in matrices. The complexity is reduced due to separation of concerns. For example, release engineer should be aware only about end products of domain PL, and with this information to form an end product for vehicle. He does not need to know about all the implementation details of feature declarations, which are important for feature owner for instance.

<table>
<thead>
<tr>
<th>Structures and tactics</th>
<th>GEARS Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi PL</td>
<td>Nested Production Lines</td>
</tr>
<tr>
<td>Modular PL</td>
<td>FM in Mixins and Assets</td>
</tr>
<tr>
<td>Context variability model</td>
<td>Mixin in the top level production line</td>
</tr>
<tr>
<td>FORM Feature Categories</td>
<td>Feature groups</td>
</tr>
<tr>
<td>Asset types</td>
<td>System and Software Assets</td>
</tr>
</tbody>
</table>

Table 28 Technology implementation of selected approaches

Context Variability model is implemented as a mixin in the top level product line. Dependencies can be established between Context Mixin and other production lines. Also Context Variability mixin can be imported in the lower level production lines to make assertions possible. Importing a mixin still satisfies the requirement of “no redundancy” as importing is done by reference. The location of the mixin remains single. To cope with all kinds of inconsistencies, profile values as <inherit> should be used. This key word means that mixin’s profile value is taken from the higher level product line. <inherit> should not only be used with Context Variability mixin, but all the other mixin modules that are used in the several locations.

As mentioned in Section 7.3.4 decomposition can be made according to asset types in the system architecture. That means existence of different feature models for requirements specification, design components, test artifacts. In GEARS this tactic is achieved by using mixins to model common parts of feature models and later assemble them into GEARS assets. GEARS assets can be linked with the help of bridges to IBM DOORS modules, Rhapsody components, Rational Quality Manager test specifications. Separate variability models can be defined in assets (Figure 33).
To maintain traceability between mixins, feature declaration and product lines the dependencies should be defined. GEARS operate with the concept of assertions, which are used as constraints and also requires/excludes relations. Four different dependency types that are needed to implement feature modeling in the case organization were identified. Types and GEARS implementation are showed in Table 29.

As the system is decomposed to multiple product lines that are nested inside each other there is a need for defined or undefined products in the matrix. Composition Assertion allows to specify when the imported production line is required to be defined or undefined in a Matrix product. The architecture may require that some components be used in combination or that the use of one component precludes the use of another. These dependencies are described through composition assertion statements.

When one product line consists of several mixins, these mixins can also be dependable. Features declared in different mixins may require a relation to ensure that some condition is always held. Product Line Assertions may reference any feature declared in the Product Line’s Mixins.

The other situation is when features declared in mixins from different product lines need to be constrained. To make features visible inside another product line, first it should be imported by reference. After that the product line assertions can be defined, as in the previous case.

The most common dependencies are between feature declarations in one feature model (mixin). To define these constraints mixin Assertions should be defined.

<table>
<thead>
<tr>
<th>Dependency</th>
<th>GEARS Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Between Nested PL</td>
<td>Composition Assertion</td>
</tr>
<tr>
<td>Between Feature Models in the same Product Line</td>
<td>Production Line Assertion</td>
</tr>
<tr>
<td>FM in PL1 to FM in PL2</td>
<td>Import (by reference)FM to PL1, define Production Line Assertion</td>
</tr>
<tr>
<td>Feature to Feature in FM</td>
<td>Mixin Assertions</td>
</tr>
</tbody>
</table>

Table 29 Dependency types and GEARS Implementations

However, there are still some challenges that are presented in Table 30.
<table>
<thead>
<tr>
<th>Challenge</th>
<th>Possible Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>No direct links between decomposed modules</td>
<td>Traceability possible only by nesting product lines. Naming convention might be useful</td>
</tr>
<tr>
<td>Modeling cross cutting features</td>
<td>Identify main domain of the feature and import mixin of other domains used</td>
</tr>
<tr>
<td>No feature cardinality (e.g. select exactly 2 variants out of 3 possible)</td>
<td>Create boolean assertion (but the expression will get too big)</td>
</tr>
<tr>
<td>Ensure consistency</td>
<td>Use &lt;inherit&gt; attribute of the profile</td>
</tr>
<tr>
<td>Defining Views</td>
<td>Create separate feature models</td>
</tr>
</tbody>
</table>

Table 30 Challenges in implementation

7.4.3 Park Assist Example and Lessons Learned

This section provides the results of modeling of the Park Assist customer feature with GEARS. The models are not complete and were used for illustration and communication purposes with company’s practitioners. Due to time restrictions of practitioners the modeling of the full park assist is only being started. That is why there is no much feedback on the details of the approach. However, during preliminary modeling concepts explained in previous sections were discovered and used. Moreover, some patterns of ambiguities that should be avoided were gathered.

Parking Assist (PA) is a customer visible feature in the vehicle that helps the driver in parking the car. PA can be provided in several variants, from the standard one to the advanced where the car is parked automatically. Based on the provided functionality the PA profiles can be defined. Figure 34 shows two types of parking: in the parallel slot and perpendicular slot. The other functions that vary are the type of feedback (audio, video, and hatchback), the type of guidance (instructions, automatic steering or automatic gears shift), and the type of searching for a free parking slot.

Figure 34 Parking Assist

7.4.3.1 Modeling Infrastructure

To model this feature we first needed to establish the initial product lines structure. Product lines of related domains and functions were created. Error! Reference source not found. Figure 35 shows the structure of Vehicle product line. Vehicle product line is a top level product line that consists of VehicleOptions mixin and imports product lines of domains. In this implementation ActiveSafety domain, Entertainment domain and PowerTrain domains are presented.
Vehicle Product Line defines valid product in the matrix. The mixins and imported product lines are instantiated with available profiles. Figure 36 shows two defined products:

- Car for European Market, where Vehicle Options are set to EU and ActiveSafety products are set to Advanced (“AdvancedActiveSafety”)
- Car for USA Market, where Vehicle Options are set to USA and ActiveSafety products are set to Simple (“ActiveSafety”).

As mentioned before vehicle option codes are modeled as context variability model by the means of mixin on the top level PL. Figure 37 shows the Structure view of VehicleOptions and also two profiles (EU and USA) in the left column. Feature declarations modeled are Driving Side, Country, Brand and Model designators, Body Style and Car Line.
Active Safety domain is a product line by itself. It imports product lines of customer visible features, defines features that are common for several nested product lines and specifies valid end product in the matrix. Figure 38 shows how active safety domain can be structured.

Object Detection feature can be used in Parking Assist and also in some other features (as cruise control for example). That is why it was modeled as a mixin in the domain product line. Figure 39 pictures feature declarations of the Object_Detection feature.
As for all the other product lines in GEARS, the matrices with products should be defined. So, matrix “product” was created for the active safety domain. Active safety package can come as Advanced or Standard configuration (Figure 40). Profiles for Object_Detection and ParkAssist are selected.

In this section it can be seen how product lines, mixins and matrices are the building blocks of the top level production line. They can be assembled together or used independently. Mixins can be reused within product lines and assets.

7.4.3.2 Modeling of Park Assist Features

Previous section described the initiating of product line structure. This section concentrates on the parking assistance feature.

Park Assist is modeled as a product line that is imported into Active Safety domain product line. Figure 38 also presents the mixins of park assist product line. First of all, park assist feature declarations depend on some of the VOC or global parameters as Transmission Type. That is why VehicleOptions are imported into mixins of ParkAssist product line. The same situation happens with ObjectDetection. Park Assist requires ObjectDetection functionality when the “Search” is enabled and to detect the measurements of the free spot. To implement these dependencies the mixins should be imported by reference.
The first attempt to model discussed feature by practitioner before the introduction of structures and tactics is presented in Figure 41 on the structure view. As can be seen functions were separated in three groups: functions, miscellaneous and HW_Components. This structuring is ambiguous and redundant.

Figure 41 First model for Park Assist

The following attempts brought more structure to the model, but redundancy was still not avoided. Functionality was grouped into functions, vehicle options, sensors and hardware implementation. Vehicle options were later moved into the top level product line. Sensors and hardware information can be moved into separate mixin as well, and if needed used within an asset.

The next two figures present two variants of modeling PA, with feature groups and without. The decision on which style to use depends on practitioners. Figure 42 shows the structured view of the feature model that shows no feature groups. Only functionality features that present variability between products are being modeled. Figure 43 shows the same feature but by using feature categories. The graphical view is shown, but is wrapped up for better readability. We can also notice that Operation Environment feature group consists of weather conditions and road conditions. By applying tactics of evolution, we can predict that these properties can be used by other features as well. So, they need to be moved to the top level product line.
Figure 42 PA feature model
After specification of feature declarations feature profiles and product were defined. Products of parking assist are presented in Figure 44. Notice, that VehicleOptions and Object_Detection mixins got `<inherit>` value to ensure that consistency between different product lines.

### 7.4.3.3 Lessons Learned

During modeling some interesting aspects about defining constraints, reduction of complexity and avoiding ambiguities were learned. They are presented in the following list.

1. There are a big number of dependencies. Modeling and discovering them takes time

2. Sometimes there are several possible decompositions of features that are semantically correct (e.g. Figure 45). Both decompositions are semantically correct and do not contradict specification. If both variants are possible, consider understandability issues (e.g. when number of children is big, consider grouping). Consider possible changes and evolution.

3. Avoid redundant feature declarations. Consider Figure 46. Feature “Perform Parking” is constrained with the existence of “ElectronicPowerSteering”. To
model this dependency “PerformParking” children should be related to “PowerSteering” feature (that might be in other model). Boolean Feature declaration “ElectronicPowerSteeringNeeded” is redundant, as the same information is already represented by assertion. However, this decomposition is easier for understanding dependencies to other modules.

![Figure 46 Avoiding redundancy](image)

4. Look out for semantic difference in the structures. Some structures may seem correct at the first glance, but in the precise analysis they might have different meaning.

### 7.5 Discussion and Conclusions

This chapter discussed the possibilities of applying feature modeling structures and tactics in the context of case organization. Through workshops, modeling and brainstorming the approach that fit the organization needs and is compliant with GEARS tool chain was selected. It is defined that functional decomposition will best fit the initial needs of the organization. Such supporting techniques as asset types, views, feature categories and context variability model were also followed.

GEARS main concepts were presented and described so that the reader can understand the implementation techniques. GEARS specific approach to structuring modeling space was described.

However, the results of modeling are only preliminary. But the work in this direction is being done in the organization.

The results of this chapter can not only used for the case organization, but also considered by other practitioners. Of course, other organizations may have different problems and concerns. That is why benefits and challenges of each approach were discussed. Within the different context and tool, the set of approaches to follow might be different. Nevertheless, the information presented should be useful in making a reasoned decision.

Next chapter discusses if the goals stated in the beginning of the thesis are achieved by the process selected.
8 Evaluation of Approach with GQM

In Section 5.3.2 “Improvement Goals. GQM” the measurable goals of stakeholders with respect to feature modeling were defined. These goals dealt with multiple aspects of variability management: approach, process, tool support. With defined questions and metrics the results of approach adoption can be evaluated and compared. Some metrics were not possible to collect, as they require either implemented subsystem or practitioners available to take part in some kind of experiment. Nevertheless, some improvement tendencies can be recognized by adopting the proposed feature modeling strategies.

The results of the evaluation are subjective, as improvement can be evaluated based only on assumptions and on the approach construction facts. The precise measurements should be taken during introduction of the selected approach, but we expect the improvement of the following quality characteristics.

Following sections will show how introduction feature modeling with GEARS helps to improve the stated goals. The questions posed for GQM template are being answered. The answers were discussed in the case organization with practitioner. In general, the posed goals were achieved as stated by the company’s practitioners. The approach was presented in front of the NGT working group and got a positive feedback.

The numbering of goals and questions correspond to the one in the Section 5.3.2.

8.1 G1 Understandability

The first goal that was defined is to improve understandability of variability information in the system. By answering the following questions we can interpret if the goal was achieved.

Q1. Is specification of product variants done easier than before? The answer to this question is rather subjective, as it relies on the practitioner’s rating. To be able to rate, they first should try to model with the new approach. So, the exact answer to this question cannot be given. However, provided clear decomposition tactics and tools should make the life of feature owners easier.

Q2. How efficient is specification of product variability? To measure efficiency of specification the experiment should be set up to take the time needed to model the feature. So, answer to this question is undefined.

Q3. In what artifacts is variability information described? After the adoption of feature modeling variability information is consolidated in one place. It is described in feature models and profiles in GEARS tools. Information is not only in the single place, but is in the structured form. Before it was spread throughout the artifacts – requirements specifications, use cases, additional documents.

Q4. In how many artifacts is variability described? Today variability is consolidated in one centralized place (GEARS).

Q5. What is the time required to understand product variants? The exact time that is needed to understand product variants is unknown. However, the functionality that is provided by the selected tool allows fast (at least faster than before) overview of what products are available and what are their differences. This is achieved by profiles and matrix combination.

Q6. Can features and variations be viewed at different level of abstraction? The models are structured in the way that allows creation of feature models for different abstraction levels. Through matrices only end product profiles can be seen. However, the automatic change of abstraction is not possible. The answer is – partly satisfied.
Q7. Can different views of the feature model be defined? The mechanism of views is not supported by GEARS tool. However, by construction, we wanted to achieve that feature models can be reused to create asset variability models that are might be relevant only to specific user group. To some extend the requirement is satisfied by construction.

Q8. How much time is needed to understand dependencies between features? The value is unknown. But we assume that the time required will be less than before introduction of the feature models. The dependencies between features are explicitly modeled in the form of assertions. Dependency types and their representation are described in the previous chapter. This helps in understanding the dependencies.

Q9. Does the approach describe possible dependencies between features and the way to express them? The approach defines dependencies of four types and the way to define them. Also the GEARS tool provides mechanism of assertion that allows creating constraints based on Boolean logic.

The answers to the posed questions can be interpreted in the positive way. The defined decomposition strategy, dependency modeling, consolidating variability information in one place altogether works in favor of improving understandability.

8.2 G2 Correctness

The correctness of the specification of variability information cannot be assured automatically. The quality review process should be established and followed to assure correctness of modeling and assembling of products. The questions for G2 are answered bellow.

Q10. Does feature model contradict the requirement specification documents? This question should only be answered after the modeling is done. But we can assume that defining rules of decomposition that correspond to structure of requirement specification will help to avoid confusion. This is only an assumption, and as we have seen, semantic mistakes can happen.

Q11. Can all valid products be described? By construction all valid products can be described. Any number of products can be assembled from profiles of feature declarations.

Q12. Is it possible to define a configuration that contradicts specification? If all the constraints between features are modeled than any configuration that contradicts these requirements is not able to create.

Q13. Is naming of features unambiguous? The naming conventions were not taken.

As can be seen, there are all possibilities to prevent mistakes from happening, but correctness still depends on the person who is doing modeling.

8.3 G3 Consistency

Consistency can be in the big extent ensured by construction. Following questions for G3 are being answered.

Q14. Can the variability information be traceable between different artifacts? With the use of bridges between GEARS and other systems variability can be traced from feature models to requirement specifications, design artifacts, test specifications.

Q15. Is traceability ensured by tool support? Yes.

Q16. Can the variability information be traceable between different levels of abstraction? The refinement relationship is not defined by tool. Traceability only in the form of REQUIRES relationship can be established. This type of link is not realized within tool.
Q17. Is the single notation used? The single notation for feature modeling is used.

Q18. Are similar situations addressed similarly? This thesis defined the decomposition tactics that should be followed. If the feature owners adhere to the rules, then the situations will be addressed similarly. The formal metric for this question was the ratio between number of identified patterns and number of different solutions found. This should be analyzed by quality review department after modeling is complete.

From the five questions, four can be answered positively. This can be interpreted as an improvement compared to situation before the feature modeling and creation of guidelines started.

8.4 G4 Completeness

The other goal was the improvement of completeness with respect to variability information. To recall, the variability information was not structured and dependencies not modeled explicitly. Nevertheless, we need to at least show that the information is as complete as it was before the approach was introduced.

Q19. How many feature variants are described in Requirements Specification?
Q20. How many feature variants are modeled?

The ratio between Q19 and Q20 should give the exact number on how complete is specified variability. On the other side, approach and tool support provide all the means to make it as complete as possible.

Q21. Is it possible to model all the dependencies? It is possible by construction.
Q22. Are all the dependencies between features modeled? Completeness of modeled dependencies should be reviewed by the quality department.

Based on the posed question we can say that completeness, as correctness, depends on modelers and should be regularly reviewed. Selected approach and tools, support the modeling tasks in the way to make possible to model all variants and all the dependencies.

8.5 G5 Modularity

The proposed decomposition approach concentrates on the quality of modularity. By construction, system is decomposed into product lines and modular feature models. If the modularity is better than before cannot be evaluated, as there is no baseline defined.

Q23. Is impact of the change local? To answer the question following information should be gathered:

- number of affected features, assertions, and dependencies
- number of affected artifacts
- number of affected models

The values are not available, but could be gathered in the future, when the models are established. However, the construction of the approach aims to better modularity and minimization of the change impact.

8.6 G6 Redundancy Reduction

The approach defines the concept of creating the context variability model with all the global variations. Feature declarations should be imported to other product lines, in case they are referenced from more than one PL or asset. The tool allows access to these features by
reference from any other model or product line. This assures that no information is reused by copying. The question asked was Q24.

Q24. Are there copied pieces of feature model? If the advice is followed then there should not be copied information. By construction, features should be reused only by reference.

Another two relevant questions to help interpret the results are Q17 and Q18, that were previously discussed.

Q17. Is the single notation used? The single notation for feature modeling is used.

Q18. Are similar situations addressed similarly? This thesis defined the decomposition tactics that should be followed. If the feature owners adhere to the rules, then the situations will be addressed similarly. The formal metric for this question was the ratio between number of identified patterns and number of different solutions found. This should be analyzed by quality review department after modeling is complete.

Based on the positive answers it can be stated that by the approach construction the redundancy is avoided.

8.7 G7 Maintainability

During the selection of structures and tactics the maintainability issue was taken into consideration. First of all monolithic feature model was discarer, because of too much complexity and impossibility of maintaining. Another point is reduction of redundancy, which was discussed before (questions Q17, Q18, Q24). It is also obvious, that better understandability also favors into better maintainability. The unique questions for goal G7 are:

Q25. Can the patterns of unnecessary complexity be defined? The approach should define the complexity types, in the form of situations and patterns. Few situations were defined in this work. However, more should be added and gathered during the modeling itself.

Q26. Are the defined complexities avoided? This question cannot be exactly answered at this point of time.

Q27. Can model be used and developed in parallel? The decomposition of system into multi product lines and mixins supports the parallel work and separation of concerns.

The improvement tendency of maintainability of variability information can be noticed. The construction of the approach took into consideration this issue. Of course, the baseline should be established; measurements should be taken and compared to evaluate improvement objectively.

8.8 Summary

Table 31 shows the summary of fulfillment of stated GQM goals by the feature modeling approach that was synthesized for the case organization. The answers to the questions discussed in this section were used to reason if the subgoals and thus goals were satisfied or not. We can see that the goals are satisfied or partly satisfied.

Tool support is very important for both successful usage of the approach and conducting realistic measurements. Most of the requirements are addressed by the approach. However, still good tool support is needed. In this work GEARS tool were used. Nevertheless, another tool can be selected, but it should support the defined requirements. The combination by the structuring approaches, tactics and comfortable tool support is mandatory for a successful implementation of feature modeling to the big systems.
<table>
<thead>
<tr>
<th>Goal</th>
<th>Subgoal</th>
<th>Satisfied</th>
</tr>
</thead>
<tbody>
<tr>
<td>Understandability</td>
<td>S1. Specification of product variants is done easier than before</td>
<td>+</td>
</tr>
<tr>
<td></td>
<td>S2. Understanding of product variants is done easier than before</td>
<td>+</td>
</tr>
<tr>
<td></td>
<td>S3. Comprehensive overview of the high level features and their variants</td>
<td>+</td>
</tr>
<tr>
<td></td>
<td>S4. Defined view according to point of interest</td>
<td>+/-</td>
</tr>
<tr>
<td></td>
<td>S5. Dependencies between features and their variants can be analyzed</td>
<td>+</td>
</tr>
<tr>
<td>Correctness</td>
<td>S6. Model should not contradict requirement specification (reality)</td>
<td>+</td>
</tr>
<tr>
<td></td>
<td>S7. Created Product Configurations should be a valid product</td>
<td>+</td>
</tr>
<tr>
<td></td>
<td>S8. Specification of requirement variants is unambiguous (There is only 1 way to interpret variability in FM)</td>
<td>+/-</td>
</tr>
<tr>
<td>Consistency</td>
<td>S9. Variable information should be traceable between different artifacts</td>
<td>+</td>
</tr>
<tr>
<td></td>
<td>S10. Variable information should be traceable between different levels of abstraction</td>
<td>+</td>
</tr>
<tr>
<td></td>
<td>S11. Variability information should be described in the uniform way</td>
<td>+</td>
</tr>
<tr>
<td>Completeness</td>
<td>S12. All possible variants of the feature should be modeled</td>
<td>+</td>
</tr>
<tr>
<td></td>
<td>S13. All dependencies between features should be modeled</td>
<td>+</td>
</tr>
<tr>
<td>Modularity</td>
<td>S14. Minimize impact of the change</td>
<td>+</td>
</tr>
<tr>
<td>Decrease redundancy</td>
<td>S15 (S11). Variability information is specified in the unified way</td>
<td>+</td>
</tr>
<tr>
<td></td>
<td>S16. Variability information should not be duplicated</td>
<td>+</td>
</tr>
<tr>
<td>Maintainability</td>
<td>S17. (S11) Variability information is specified in the unified way</td>
<td>+</td>
</tr>
<tr>
<td></td>
<td>S18. (S16) Variability information should not be duplicated</td>
<td>+</td>
</tr>
<tr>
<td></td>
<td>S19. Unnecessary complexity should be avoided</td>
<td>+/-</td>
</tr>
<tr>
<td></td>
<td>S20. Separation of work should be allowed</td>
<td>+</td>
</tr>
</tbody>
</table>

Table 31 Evaluation of approach by GQM

Table 32 shows if the requirements stated by stakeholders were fulfilled by the approach. As can be noted most of the requirements are satisfied or partly satisfied.
<table>
<thead>
<tr>
<th>Case Study Requirements</th>
<th>Supported by Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Process is supported by tool</td>
<td>+</td>
</tr>
<tr>
<td>2. Variability information is captured in one place</td>
<td>+</td>
</tr>
<tr>
<td>3. Explicit modeling of variability by feature modeling</td>
<td>+</td>
</tr>
<tr>
<td>4. Variability information is structured</td>
<td>+</td>
</tr>
<tr>
<td>5. Feature based development</td>
<td>+</td>
</tr>
<tr>
<td>6. Impact analysis is more automatic</td>
<td>+</td>
</tr>
<tr>
<td>7. Modeling guidelines should be available for the approach</td>
<td>+</td>
</tr>
<tr>
<td>8. Approach should be scalable</td>
<td>+</td>
</tr>
<tr>
<td>9. Approach should give advice on granularity of models</td>
<td>+</td>
</tr>
<tr>
<td>10. Approach should support evolution</td>
<td>+</td>
</tr>
<tr>
<td>11. Feature modeling approach should be incorporated with the whole development process</td>
<td>+</td>
</tr>
<tr>
<td>12. Modeling of only variable features</td>
<td>+</td>
</tr>
<tr>
<td>13. Modeling of excludes/requires interdependencies</td>
<td>+</td>
</tr>
<tr>
<td>14. Hierarchical inclusion of lower level feature model into higher level feature model</td>
<td>+/-</td>
</tr>
<tr>
<td>15. Graphical representation</td>
<td>+</td>
</tr>
<tr>
<td>16. Structural/text representation</td>
<td>+</td>
</tr>
<tr>
<td>17. Switch between different representations</td>
<td>+</td>
</tr>
<tr>
<td>18. Automatic instantiation of requirements specification,</td>
<td>+</td>
</tr>
<tr>
<td>documentation, test cases and so on</td>
<td></td>
</tr>
<tr>
<td>19. Configuration control of feature models</td>
<td>+/-</td>
</tr>
<tr>
<td>20. Parallel development of features</td>
<td>+</td>
</tr>
<tr>
<td>21. Model should not contradict requirement specification(reality)</td>
<td>+/-</td>
</tr>
<tr>
<td>22. Created Product Configurations should be a valid product</td>
<td>+</td>
</tr>
<tr>
<td>23. Specification of requirement variants is unambiguous (There is only 1 way to interpret variability in FM)</td>
<td>+/-</td>
</tr>
<tr>
<td>24. All possible variants of the feature should be modeled</td>
<td>+</td>
</tr>
<tr>
<td>25. All dependencies between features should be modeled</td>
<td>+</td>
</tr>
<tr>
<td>26. Minimize impact of the change</td>
<td>+/-</td>
</tr>
<tr>
<td>27. Specification of product variants are done easier than before</td>
<td>+</td>
</tr>
<tr>
<td>28. Understanding of product variants are done easier than before</td>
<td>+</td>
</tr>
<tr>
<td>29. Comprehensive overview of the high level features and their variants</td>
<td>+</td>
</tr>
<tr>
<td>30. Defined view according to point of interest</td>
<td>+/-</td>
</tr>
<tr>
<td>31. Dependencies between features and their variants can be</td>
<td>+</td>
</tr>
<tr>
<td>Case Study Requirements</td>
<td>Supported by Approach</td>
</tr>
<tr>
<td>------------------------------------------------------------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>analyzed</td>
<td></td>
</tr>
<tr>
<td>32. Variability information is specified in the unified way</td>
<td>+</td>
</tr>
<tr>
<td>33. Variability information should not be duplicated</td>
<td>+</td>
</tr>
<tr>
<td>34. Unnecessary complexity should be avoided</td>
<td>+/-</td>
</tr>
<tr>
<td>35. Separation of work should be allowed</td>
<td>+</td>
</tr>
<tr>
<td>36. Variable information should be traceable between different</td>
<td>+</td>
</tr>
<tr>
<td>artifacts</td>
<td></td>
</tr>
<tr>
<td>37. Variable information should be traceable between different</td>
<td>+</td>
</tr>
<tr>
<td>levels of abstraction</td>
<td></td>
</tr>
</tbody>
</table>

Table 32 Requirements fulfillment
9 DISCUSSION

This chapter discussed the results obtained in the previous chapters of this work. The reflection back to the chain of reasoning will be given and the implications for the industry will be shown. This chapter is completely based on the previous results of the work, but aims not to repeat already stated summaries and achievements.

The overall goal of the work for the industry was to select, create or synthesize approach for feature modeling in the case organization. The studied organization is a big automotive company in the process of introducing new tool chain for more efficient and quality engineering. The objectives of stakeholders in the beginning of the work were not clear and the requirement and expectations to the feature modeling was not understood properly. That is why, first of all the context of the organization was studied:

- current processes with its problems
- challenges in adoption of product line engineering
- expectations from the new approach

As a result the list of requirements that consists of 37 points was formulated. For proper evaluation of the new approach a set of GQM goals were set with respective questions and metrics. These results were presented in Chapter 5 and provide an answer to the RQ1.

At this point of time we knew about problems, challenges and expectations of the practitioners. To solve this problem the previous literature work was gathered on the topic. The search concentrated on the structuring and decomposition approaches. Of course, different views on the problem and solution are possible. This work looked at the problem from the perspective of organizing feature model in space and supporting this organization by tool chain. The problem might be solved by applying some FM notations or SPL frameworks, but this was not relevant in our context.

So the information on the approaches to structure large scale product lines was found. The mapping of requirements to approaches showed that there is no single remedy. The list of the tactics (decomposition according to some principle or criteria) and structures (monolithic models, modular FM and MPL) is the answer to RQ 2. This was Chapter 6.

Bringing together results of RQ1 and RQ2, knowing about the company’s context and automotive industry specifics, the solutions from Chapter 6 were researched in the industrial setting. Together with the practitioner from the company several strategies were combined to fulfill the requirements stated in the beginning of the work. The results of this work were presented in Chapter 7.

The way of structuring FMs can be one of the success factors. Consider decomposing a system that is clearly oriented of functional architecture, according to physical structure. First of all during modeling practitioners will face a number of problems:

- What is the mapping between features and physical parts
- Where the feature should be modeled
- If feature is spread across parts how it should be modeled
- And maybe many more

During maintenance the amount of rework can grow and feature modeling can become not effective.

During this case study, when approach required change in thinking and looking on the system, it was extremely hard to come up with a model even of small size. Obviously, this effort could not be accepted by the company. The decomposition tactics should have been
natural and easy for adoption of practitioners. The other example can be modeling along the evolution. This tactics, being promising, first of all requires change in people’s mindset. And this may be one of the biggest problems. If the new process is not comfortable for practitioners the adoption will be slow and have no support among people.

However, when the decomposition is natural and comfortable, problems can be avoided.

Another point was finding the right granularity of models. For instance, software related functionality of the feature and hardware related variability can be modeled inside one feature model or split up into two. Both options are valid. However, for the case organization the consistency of modeling is important. That is why the decision in favor of one of the methods should be taken based on some objective criteria. This is a found tradeoff between modularity and understandability. The improved modularity requires having two FMs, but it is easier to understand when all the variability about feature is in one model (in case it is not too big). Using the QFD prioritization from Chapter 4, better modularity is more important than understandability. However, the researcher can only advice here and leave the final decision to the company’s experts.

The results show that for such a complex domain as automotive the single decomposition tactic and structuring approach is not enough. To satisfy company’s goals several approaches are combined. Each of them focuses on one of the aspect of FM quality. For example, view based modeling satisfies the requirement of having multiple views, but lacks the qualities of evolution support, traceability and others. These qualities are just not addressed by the approach.

The synthesized approach was used to model the subsystem “Parking Assist”. This is a good example for practitioners to start with feature modeling. During interview some of the respondents felt not confident on where to start their work. Having the initial product line infrastructure should help to solve this problem. However, the infrastructure did not include configuration control and version control management of the models. But, GEARs operate with the file system, so it should be compatible with the standard version management programs.

Some goals and requirements could only be satisfied with the proper tool support. Interesting that GEARs tool fulfilled almost all of them to some extent, except but support of Views (as a filter mechanism). This could have happened because some of the respondents (2 out of 5) had GEARs training prior to interview. This could cause their bias. But, the other respondents also expressed similar expectations from the tool. Moreover, there is not much industry ready tools for SPL engineering on the market that also provide traceability bridges to the requirements management tool, design and test tools.

This work can be seen as a starting point for the industry and the results can be improved over time based on the industrial experiences.

The last but not least, is the people factor. First of all the approach should be supported by people who are going to use it. The acceptance is important as these are the people who will do the actual modeling. Some requirements could be satisfied only by correct following of the approach and guidelines. The minus is the practitioners in the company are new to FM and SPLE in general. The mistakes cannot be avoided, that is why it is recommended to have a quality department that will look after the quality of the created models. The reviews of the feature models can be also done inside the working group (as the analog of the code review). As a result of tight cooperation with the case organization the proposed solution had a very positive feedback inside the company.
10 Conclusion

This chapter gives a short summary of the results of this work and the recommendations to the future work.

10.1 Conclusions

Feature modeling is one of the techniques of variability management that helps to organize and structure the domain. It is the way to capture system’s variability in one place. Automotive domain deals with a huge number of features and functions. To make development more efficient the product line approach is being applied. However, the domain specifics do not allow following methods that were already in use. This thesis researched problems with feature modeling in automotive domain, available decompositions of modeling space, tactics that could be applied to reduce complexity of the domain. Based on available information the new approach was synthesized. The work was carried out in the context of automotive company whose product line can be referred to as large scale or mega scale.

The goal was to evaluate the suitability of available feature modeling approaches for automotive industry and to propose a guideline that can be followed to successfully capture variability. According to this goal four research questions were stated. To reach our goals the case study was performed. To decide on the approach first the situation in industry should have been characterized. This was done by means of survey where main data collection procedures were semi structured interviews and workshops. After that literature review was carried out to find out available feature modeling approaches. Results of literature study were discussed with practitioners and put into context of the case organization. That allowed us to formulate strategy to feature modeling that fits the best to the needs of organization. The methodology used to research the problem allowed to get all the information that was needed.

The answers to posed research questions were found.

RQ1. What are the challenges and requirements of feature based software and system development in automotive industry?

The root problem of automotive system is their complexity. The huge number of features makes it almost impossible to manage. The main requirements to the approach are scalability, modularity and redundancy avoidance. Not of less importance is ensuring of traceability, correctness and completeness. The detailed description of these qualities were given and formulated according to GQM.

RQ2. What hierarchical feature modeling approaches exist in literature, what are their characteristics and how are they applied in industry?

The number of approaches to modeling features exists. However, the detailed guidelines on modeling, complete subsystem examples and domain specific comparison of approaches are not yet available. This thesis discussed such structuring approaches as multi product lines, context variability model, modular feature models, and multi criteria product lines. The other direction of discussion was decomposition criteria such as, organizational structure, by asset type, reflecting solution space structure, based on marketing needs and so on. The approaches proposed are not well established and real life experience reports are not available.

RQ3. What hierarchical feature modeling approaches satisfy the requirements of automotive industry? How the respective models look like?
In this work we discussed feature modeling tactics and structures and put them in the context of case organization. We selected decomposition based on functional architecture. The structure applied is a combination of hierarchical product lines and modular feature models. Complementary techniques are establishment of context variability model and creating variability models depending on asset type.

**RQ4. How the respective management process should look like?**

The synthesized approach was implemented using GEARS tool chain that allows management of variability through the whole development lifecycle. Of the main importance in this work was the modeling part. However, we considered the possibilities provided by tool support, as traceability links between other artifacts and restriction of complexity using “profiles”. The tool’s concepts and process of modeling was described and illustrated on the example.

The selected approach and tool was evaluated according to stated GQM goals.

The process proposed in this thesis work is being introduced in the case organization. However, the implemented subsystem is still not ready due to other more prioritized work. Once, the subsystem is modeled the more precise evaluation of the approach should take place. All problems that are not mentioned in this work should be noted for further improvement of the approach.

The results of the work is different from the other studies as the work discussed and brought together different feature modeling approaches and used them to define an approach suitable in automotive industry. The work concentrated on the structuring of feature models in automotive specific context and selection of tactics to reduce complexity of the models.

### 10.2 Limitations and Future Work

This work should be continued further in the several directions. First of all the work on modeling of features by the case study organization should be continued. This work includes prior modeling of requirements, creating feature models, constrains over them and rules according to which the system should be instantiated.

The second direction is taking measurements according to defined GQM. This will provide the objective criteria for evaluation and decision taking. It is possible that during large scale modeling the problems might be discovered that were not noticed during this work. In this case, the approach should be tailored and the existing documents updated accordingly.

The third direction is creating a link between created feature models to other artifacts. During this work the links to requirements artifacts were considered. However, the goal is to incorporate feature modeling into all levels in V Model. So the next step may be linking feature models and design artifacts or test artifacts.

One of the important points is the remaining issue of scalability. Scalability was defined as one of the main problems with the current approaches to feature modeling for automotive industry. However, the stakeholders of GQM did not define it explicitly in the goals section. Nevertheless, scalability is indirectly addressed by taking into consideration easiness of modeling and usage of created feature models, establishing processes, tool support. The selection of structures of feature model was important for decomposing big systems.

The other limitation is unavailability of benchmark for comparison. By contraction of the approach some of the GQM measurements were gathered by could not be objectively compared to the “situation before the improvement”.

One of the future steps should be repeating this work in the other case organization to support reliability of the results. As stated previously in the different parts of this work, the case study was done in only one organization; therefore it cannot be objectively argued that
the results can be applicable to the other organizations and industries. Nevertheless, the research design, identified tactics and structures, reasoning for their selection or combination can be reused in the other studies and organizations.
11 REFERENCES


30. Dhungana, D. and Grünbacher, P. *Understanding Decision-Oriented Variability Modelling*.  


APPENDIX 1. INTERVIEW QUESTIONS

Interviewee: 
Interviewer: 
Date: 

The interview sessions on Variability Management (VM) are conducted to analyze the current situation in the field of VM in your organization. The first goal is to gather knowledge about current VM usages, model creation and change management. The second part of the interview focuses on challenges and possible improvements of the current processes.

The interview results will help to understand current state of affairs and should give a picture of strong and weak points, as well as desired and planned changes within the field of VM. Understanding these items is an important prerequisite for finding and fitting approaches that can support you with addressing demands in variability management. To get an overview as wide as possible the interviews are therefore carried out with stakeholders of different domains and roles.

You are not requested to answer the below questions before your interview takes place, but shall prepare you for the kind and span of points of interest. The interview will be separated in 4 groups:

- an Introduction of you and your role(s),
- the Current situation with respect to variability management,
- Challenges that need to be addressed and
- Improvements that should be made (from your perspective).

Part 1. Introduction

1.1 Please describe your current role!

1.2 What are your main tasks and responsibilities with respect to product line engineering?

1.3 Describe the experience you have in your current role?

1.4 Do or did you have other roles and tasks? What roles are/were these?

1.5 What is your working experience with Software Product Lines?

Part 2. Current situation

2.1 For what purpose you use created variability models? How do you use variability information in the organization?

2.2 What do you model?

- Commonalities
- Variability
- Interdependencies

2.3 How do you document variability today?

- spreadsheets
- custom profiles
- tabular formats
• matrix based
• text based
• Other

2.4 Who are the responsible persons/roles for the creation of the variability models, and what was the average timeframe spent to build these models?

2.5 Who are the stakeholders of FM and what are their respective roles (Who is using the models and how)?

2.6 What are the strengths of methods that you used so far to capture and evolve variability?

2.7 What artifacts describe variability today?

2.8 What are interdependencies between these artifacts?

2.9 What is the process of adding/changing/deleting features today?

2.10 What is the speed of evolution for the Variability Models? How fast are they changing?

2.11 How many new features per month you have to integrate?

2.12 How many of that features are common among products?

2.13 What is the size of feature models now?

2.14 What is the number of subtrees, modules, interdependencies?

2.15 What is the average and maximum size for model depth, branches and leaves?

2.16 “Scoping” is defined as process of identifying and bounding areas (subdomains, existing assets) and capabilities (features) of the product line where investment into reuse is economically useful and beneficial to product development.

Do you use scoping (as defined above) to analyze/assess/model your domain?

2.17 If you use scoping then how it contributes to the modeling of domain?

Part 3. Challenges

3.1 What are the deficiencies of the approaches that you used so far to model variability? (Challenges in the process of capturing and documenting variability)

3.2 What are the deficiencies of the notations that you used to model variability so far?

3.3 What are the key challenges in adopting, creating, and maintaining VMs?

3.4 Please, characterize these challenges in the term of effort

3.5 Please, prioritize challenges named in 3.3 starting with the highest priority.

3.6 What is the severity of the challenges named in 3.3? (high, medium, low)

Part 4. Improvement

4.1 For what purpose you plan to use created variability models in the future? How do you plan to use variability information in the organization?

4.2 What artifacts will describe variability in future?

4.3 What interdependencies will be between these artifacts?
4.4 What, in your opinion, can be improved in the procedures that you use to model variability?

4.5 What, in your opinion, can be improved in the notations that you used to model variability?

4.6 What will be the process of adding/changing/deleting features in the future?

4.7 What is the planned size of feature models?

4.8 What is the number of subtrees, modules, interdependencies?

4.9 What is the average and maximum size for model depth, branches and leaves?

4.10 What are the main reasons for adopting PL approach in your company?

- cost cutting
- customization
- increasing productivity
- improvement of quality
- Time cutting
## Appendix 2. Mapping of Case Study Requirements to the Approaches

The following table maps requirements to the feature modeling approach and tool to the approaches found in the literature, where “+” means the requirement is satisfied, “-” the requirement is not satisfied, “+/−” the requirement is partly satisfied and “o” cannot be evaluated because of lack of information or not relevancy for the approach.

<table>
<thead>
<tr>
<th>Requirements</th>
<th>FORM</th>
<th>MultiCriteria PL</th>
<th>Multi Level Feature Trees</th>
<th>Modular FM</th>
<th>Context VM</th>
<th>View Based</th>
<th>Hierarchical MPL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Process is supported by tool</td>
<td>+</td>
<td>-</td>
<td>+/-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2. Variability information is captured in one place</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>3. Explicit modeling of variability by feature modeling</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>4. Variability information is structured</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>5. Feature based development</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>6. Impact analysis is more automatic</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
</tr>
<tr>
<td>7. Modeling guidelines should be available for the approach</td>
<td>+/-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>8. Approach should be scalable</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+/-</td>
<td>+</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>9. Approach should give advice on granularity of models</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>10. Approach should support evolution</td>
<td>+/-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+/-</td>
<td>+</td>
</tr>
<tr>
<td>11. Feature modeling approach should be incorporated with the whole development process</td>
<td>-</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
</tr>
<tr>
<td>12. Modeling of only variable features</td>
<td>+/-</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
</tr>
<tr>
<td>13. Modeling of excludes/requirements interdependencies</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Requirements</td>
<td>FORM</td>
<td>MultiCriteria PL</td>
<td>Multi Level Feature Trees</td>
<td>Modular FM</td>
<td>Context VM</td>
<td>View Based</td>
<td>Hierarchical MPL</td>
</tr>
<tr>
<td>------------------------------------------------------------------------------</td>
<td>------</td>
<td>------------------</td>
<td>---------------------------</td>
<td>------------</td>
<td>------------</td>
<td>------------</td>
<td>------------------</td>
</tr>
<tr>
<td>14. Hierarchical inclusion of lower level feature model into higher level feature model</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>15. Graphical representation</td>
<td>+/-/</td>
<td>+/-</td>
<td>+/-</td>
<td>+</td>
<td>+</td>
<td>+/-</td>
<td>+</td>
</tr>
<tr>
<td>16. Structural/text representation</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>17. Switch between different representations</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>18. Automatic instantiation of requirements specification, documentation, test cases and so on</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>19. Configuration control of feature models</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>20. Parallel development of features</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>21. Model should not contradict requirement specification(reality)</td>
<td>+/-</td>
<td>+/-</td>
<td>+/-</td>
<td>+/-</td>
<td>+/-</td>
<td>+/-</td>
<td>+/-</td>
</tr>
<tr>
<td>22. Created Product Configurations should be a valid product</td>
<td>+/-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>23. Specification of requirement variants is unambiguous (There is only 1 way to interpret variability in FM)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>24. All possible variants of the feature should be modeled</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>25. All dependencies between features should be modeled</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>26. Minimize impact of the change</td>
<td>-</td>
<td>+</td>
<td>+/-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>27. Specification of product variants are done easier than before</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>28. Understanding of product variants are done easier than before</td>
<td>0</td>
<td>+</td>
<td>+/-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Requirements</td>
<td>FORM</td>
<td>MultiCri teria PL</td>
<td>Multi Level Feature Trees</td>
<td>Modular FM</td>
<td>Context VM</td>
<td>View Based</td>
<td>Hierarchical MPL</td>
</tr>
<tr>
<td>----------------------------------------------------------------------------</td>
<td>------</td>
<td>------------------</td>
<td>------------------------</td>
<td>------------</td>
<td>-----------</td>
<td>------------</td>
<td>------------------</td>
</tr>
<tr>
<td>29. Comprehensive overview of the high level features and their variants</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>30. Defined view according to point of interest</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>31. Dependencies between features and their variants can be analyzed</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>32. Variability information is specified in the unified way</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>33. Variability information should not be duplicated</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>34. Unnecessary complexity should be avoided</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>35. Separation of work should be allowed</td>
<td>+/-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>36. Variable information should be traceable between different artifacts</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>37. Variable information should be traceable between different levels of abstraction</td>
<td>+</td>
<td>+/-</td>
<td>+/-</td>
<td>+</td>
<td>+/-</td>
<td>+/-</td>
<td>+</td>
</tr>
</tbody>
</table>