A Systematic Review of Domain Analysis Solutions for Product Lines

Mahvish Khurum, Tony Gorschek

Blekinge Institute of Technology, Department of Systems and Software Engineering, S-372 25, Ronneby, Sweden

mkm@bth.se, tgo@bth.se.

ABSTRACT

Domain analysis is crucial and central to software product line engineering (SPLE) as it is one of the main instruments to decide what to include in a product and how it should fit in to the overall software product line. For this reason many domain analysis solutions have been proposed both by researchers and industry practitioners. Domain analysis comprises various modeling and scoping activities. This paper presents a systematic review of all the domain analysis solutions presented until 2007. The goal of the review is to analyze the level of empirical application and/or validation of the proposed solutions with the purpose of mapping maturity, as well as to what extent proposed solutions might be proven in terms of usability and usefulness. The finding of this review indicates that, although many new domain analysis solutions for software product lines have been proposed over the years, the absence of qualitative and quantitative results from empirical application and/or validation makes it hard to evaluate the potential of proposed solutions with respect to their usability and/or usefulness for industry adoption. The detailed results of the systematic review can be used by individual researchers to see large gaps in research that give opportunities for future work, and from a general research perspective lessons can be learned from the absence of validation as well as from good examples presented. From an industry practitioner view, the results can be used to gauge to what extent solutions have been applied and/or validated and in what manner, both valuable as input prior to industry adoption of a domain analysis solution.

Keywords

Systematic review, domain analysis, domain modeling, domain scoping, empirical evidence, usability, usefulness

1. INTRODUCTION

Software product families have received significant attention from the software engineering community since the 1990s ((Clements and Northop, 2001);(Deelstra, et al., 2004)) The concept of product lines aims towards having a set of systems that share a common, managed set of features, which satisfy the particular needs of a market segment, developed from a common set of core assets in a certain given way (Clements and Northop, 2001). The product line approach is recognized as a successful approach for reuse in software development (Kim, et al., 2007) with the major benefits of product lines adoption reported as reduced time to market ((William, et al., 2006); (Dager, 2000)), reduced cost (Pohl, et al., 2005) and improved quality ((William, et al., 2006); (Pohl, et al., 2005); (Staples and Hill, 2004)). For these reasons many companies developing software intensive products have either adopted or are considering the adoption of a software product line approach ((Bockle, 2000);(Dager, 2000)).

In order to properly introduce software product lines in a company, it is important to start with the product line domain analysis. Domain analysis can be defined as "*the process by which information used in developing software systems within the domain is identified, captured, and organized with the purpose of making it reusable (to create assets) when building new products*" (America, et al., 2001). This process can be used to identify commonality and variability in requirements and capture decisions on the ranges and interdependencies of variability. If domain analysis is not properly carried out, and ends up in defining an either too broad or too restrictive product line scope, the major benefits like reuse, cost reduction and improved quality cannot be realized (Clements and Northop, 2001).

Several domain analysis solutions for software product lines have been presented in academia and as industry experience reports. However, in order to gauge the usability and usefulness of the proposed solutions, it is important to see the empirical evidence of their application and/or validation, e.g. in industry or through experiments or tests. Furthermore, awareness has increased in the software engineering community about the importance of empirical studies to develop or improve processes, methods and tools for software development and maintenance (Sjoeberg, et al., 2005). This paper presents a systematic review conducted on the studies, which either proposed or reported on experience with domain analysis solutions or parts of it (e.g. feature modeling, commonality and variability analysis, scoping and so on), presented between the years 1998 to 2007.

The motivation was to gauge the level of actual industry adoption, i.e. to what extent the presented solutions are applied and/or validated in industry. In addition to industry validation, all other types of empirical results are collected to offer a detailed summation of the empirical evidence available. To achieve this, the selected papers are categorized and analyzed from several perspectives, such as research basis, application/validation method, level of validation and type of empirical results in relation to usability and usefulness of the proposed solutions. For industry practitioners looking to adopt a domain analysis solution the results of the study can be used as an indication of maturity as well as to estimate potential risk of adopting a certain solution. From an academic point of view researchers planning studies and evaluation of a solution can use this study as an inspiration for study design because the evaluation criteria of the review presented in this paper could be seen as a checklist to ascertain usability and usefulness.

The remainder of the paper is structured as follows. Section 2 describes the background and related work. Section 3 presents the research questions and design details of the review. Section 0 contains the results of the review as well as the categorization of the studies. In Section 4.2 results of data extraction and a detailed analysis are presented in relation to the research questions posed in Section 3.1. Section 6 concludes the paper.

2. BACKGROUND AND RELATED WORK

In this section the purpose of domain analysis activities related to domain analysis are introduced. The purpose of this is twofold, one, to provide a background to the concepts relevant for this systematic review, and, two, to describe the scope of the study.

Domain analysis is the first phase of domain engineering. The purpose of domain analysis is to gather and organize the information that is required for the smooth flow in the subsequent phases of domain engineering e.g. domain design. Domain analysis helps in the identification of the specification of the systems in the product family. Domain analysis involves various activities which can be categorized as modeling and scoping (America, et al., 2001). Modeling is defined as capturing information and organizing it into a model whereas scoping is defined as a decision-making activity.

In the modeling category, the activities identified are (America, et al., 2001):

1. Conceptual modeling contains a set of activities which identify, define, and organize the concepts relevant to the domain and their mutual relationships, to assist in formulating a precise and concise description of the domain. Information modeling is an important part of conceptual modeling.
2. Requirements' modeling contains a set of activities that capture the functional and architecturally relevant requirements for the product line and their inter-dependencies. This may also include the mapping of specific constraints to requirements.
3. Commonality and variability modeling comprises a set of activities which identify similarities and differences between the requirements. This includes the distinction of requirements that are valid for the whole domain from those that are only valid in special cases, e.g., for a specific product variant. This activity is strongly connected to domain and feature modeling.
4. Domain modeling comprises a set of activities that specify the domains and their inter-dependencies.
5. Feature modeling comprises a set of activities which identify, study, and describe features appropriate in a given domain. The objective of feature modeling is to express relations between features, properties of features, and/or superstructures of features e.g. a commonality and variability view. One of the important purposes of feature modeling is to help structure the requirements and define the allowed variants in a product line.
6. Scenario or use-case modeling comprises a set of activities which describe and model run-time behavior of members of the system family. This not only includes the functionality of the systems and their interactions with users, but also aspects such as security, safety, reliability, and performance.

In the scoping category, we find the following activities:

1. Domain scoping is the process of identifying appropriate boundaries for a domain which is appropriate for implementing systems in the product line.
2. Product line scoping is the process of systematically developing a product portfolio definition, which identifies the specific requirements and the individual products that should be part of the product line. Scope binds a product line by defining the behaviors that are "in" and the behaviors that are "out" of the product line's scope (Clements and Northop, 2001). The result of a scoping activity is a scope definition document which becomes a product line core

asset. The scope definition points out the entities with which the products in the product line will interact (that is, the product line context), and it also establishes the commonality and defines the variability of the product line (Clements and Northop, 2001).

3. Asset scoping identifies the various elements that should be reusable, i.e., the specific assets that should be part of the reuse infrastructure (core assets) as opposed to being developed application specific.

A specific domain analysis solution may not mention all these activities or distinguish between them explicitly; however it is important that these activities are discussed in relation to domain analysis. Moreover, depending on the context in which a product line is being developed some of the activities might not be relevant e.g. when only very few individual domains can be distinguished, the domain modeling activity can be omitted (America, et al., 2001). Domain analysis describes the characteristics of a class of systems, and not a specific system, and the scope will apply equally to existing products and products that have yet to be defined and built. Domain analysis can occur in a variety of contexts other than “start from scratch” product lines. For example, an organization may choose to apply the product line concept to only a part of the product portfolio.

Four previous studies ((Ramesh, et al., 2004); (Glass, et al., 2002); (Perry, et al., 2000) and (Mendes, 2005)) have reviewed the state of empirical research in different areas e.g. computer science, software engineering, web engineering and so on. However, to the best of our knowledge no other study has been conducted with the same focus as the review presented in this paper. This review does not aim to systematically classify proposed domain analysis solutions as methods, models, tools, framework or classify the studies according to the classification and evaluation scheme suggested in (Wieringa, et al., 2005). The goal of this review is to analyze practical application and validation of proposed domain analysis solutions in industry to gauge their practical usability and usefulness. In addition to this other empirical evidence is also considered e.g. proof of usability and/or usefulness demonstrated through a controlled experiment or other type of validation.

3. DESIGN

This section gives a detailed description of the review design; a definition of terms used, and discusses the validity of the study.

3.1 Research Questions and Definitions

The four research questions driving the systematic review can be viewed in Table 1.

Table 1 – Research questions and motivation.

Research Questions	Motivation
RQ1. Are proposed solutions based on needs identified in industry?	Is the solution presented based on any need/issue/problem (called *need* from here onwards) identified in industry though empirical investigation? Examples can be process assessments, case studies, participation knowledge, surveys, observations, and so on. Both direct and indirect sources will be considered, giving the presented studies the benefit of the doubt (i.e. any indication of industry basis will be considered and accepted).
RQ2. Are proposed solutions applied and/or validated in a laboratory setting or in industry?	Is the solution presented applied/validated through e.g. a controlled experiment or in industry as a part of the paper? Any validation in industry from static validation to dynamic validation will be considered, see Gorschek et al. for details (Gorschek, et al., 2006).
Research Questions	**Motivation**
RQ3. Are the proposed solutions usable?	If the authors of a proposed solution or experience report have applied/validated it in industry or demonstrated its application/validation through a controlled experiment, identify any indications/proof/reports on its usability in the venue of the application/validation (usability is defined below).
RQ4. Are the proposed solutions useful?	If the authors of a proposed solution or experience report have applied/validated it in industry or demonstrated its application/validation through a controlled experiment, identify any indications/proof/reports on its usefulness in the venue of the application/validation (usefulness is defined below).

The two main terms used in the research questions, namely *usability* and *usefulness*, are defined (by the authors) and exemplified below.

Usability, as can be seen in Figure 1 is defined in terms of:

- Scalability of Introduction. How scalable the proposed solution is in terms of its introduction cost including e.g. training, manuals and material, tools, pilot run, and tailoring to the organization in question?

- Scalability of Use. How scalable is the proposed solution in terms of its inputs, processing time and outputs? For example, if a feature modeling approach is proposed, can it handle industry grade problems, say a thousand requirements, or does it solve simplified problems with simple cases or is there any indication that industry grade scalability is possible (or even considered/mentioned/discussed by the creators of the solution)?

Scalability of Introduction and Scalability of Use point to a micro quality of a solution and that is its *efficiency*. If a proposed solution is demonstrated to have any of these aspects of efficiency, the corresponding paper is counted as having some proof of usability.

Usefulness, as can be seen in Figure 1 is defined in terms of:

- Better Alternative Investment. For example, a proposed solution (X) is better than an alternative (maybe previously used) solution (Y), and/or,
- Effectiveness. The effectiveness of a proposed solution in relation to achieving goals or solving the problems it was designed for. For example, solution X reduces time-to-market by 15%.

Again, a solution demonstrating either of the two aspects of usefulness is counted as having some proof of usefulness.

3.2 Search Strategy Development

The systematic review was performed following guidelines proposed by Kitchenham in (Kitchenham, 2007). As shown in Figure 1, a three phase search strategy was devised. In Phase 1: SPLC conference proceedings from the 2000 up to 2007 were planned to be manually searched. This was planned for several reasons. First, SPLC is the premier forum for practitioners, researchers, and educators presenting and discussing experiences, ideas, innovations, as well as challenges in the area of software product lines. SPLC also has a relatively large industry presence. Second domain analysis is a very important field and a regularly featured sub-area to software product line engineering, for the purpose of this review domain analysis solutions in relation to product lines was of primary interest. Third, since industry representation at SPLC is fairly high this includes a large amount of industry experience reports, and as one of the main features of the review is to evaluate the level of application and/or validation of the solutions, a large amount of industry experience reports was considered positive. Fourth, through manual scanning a number of keywords, alternate terms and synonyms were identified:

Population: software product lines, software product family

Intervention: requirements, requirements engineering, conceptual model, requirements model, commonality and variability model, domain model, feature model, scenario model, commonality analysis, variability analysis, domain evaluation, domain scope, asset scope

Comparison intervention: not applicable as the research questions are not aimed at making a comparison. The outcomes of our interest: the level of application/validation of the proposed solutions and their usability and usefulness evidence.

Outcomes: the level of application/validation of the proposed solutions and their usability and usefulness evidence

Out of scope: domain design, domain engineering and concepts related to architecture, implementation aspects.

In terms of context and experimental design, no restrictions are enforced.

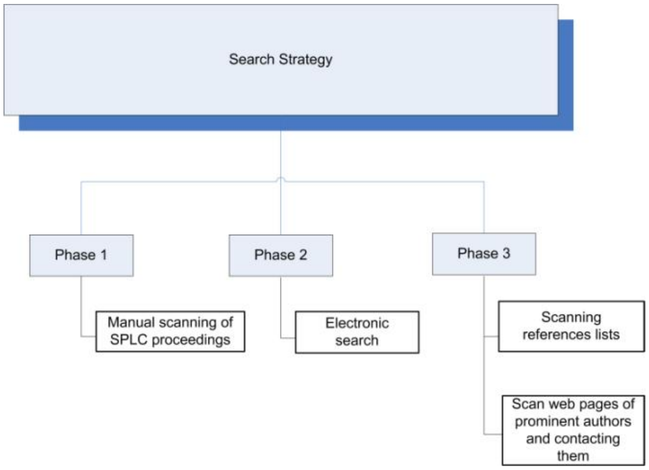

Figure 1 – 3 phase search strategy

To make the search exhaustive, in Phase 2 electronic databases were searched using the search terms deduced from the population, intervention and outcomes with the use of Boolean OR to join alternate terms and synonyms and use of Boolean AND to join major terms (Population **AND** Intervention **AND** outcomes). The electronic databases searched were:

1. Compendex
2. Inspec
3. IEEXplore
4. ACM digital library
5. ISI Web of Science

In order to ensure that search strings are comprehensive and precise, an expert librarian was consulted. The detail search strings are given in Appendix A.

Activities in Phase 3 were planned to ensure that any important research studies are not missed. Reference lists of the primary studies were scanned. The web pages of the authors in the particular area were also scanned.

Excluded from the search were editorials, prefaces, article summaries, interviews, news, reviews, correspondence, discussions, comments, reader's letters and summaries of tutorials, workshops, symposium, panels, and poster sessions.

3.3 Review Design

In this section, the systematic review design is presented describing paper identification method, inclusion/exclusion criteria, and the classification scheme.

3.3.1 Identification of Studies

Phase1: There was a total of 192 studies published in SPLC for the years 2000 to 2007, and 24 of these studies were selected after reading titles and abstracts.

Phase 2: Phase 2 had 4 steps. In Step1 (see Figure 2), 843 citations were retrieved. In Step 2 the duplicates were removed leaving 629 unduplicated citations. For all 629 citations the source of each citation, our retrieval decision, retrieval status, and eligibility decision were recorded.

In Step 3, the primary author went through all the titles to judge their relevance to the systematic review being performed. The studies whose titles were clearly not related to software product lines and domain analysis activities were excluded. For example, since our search string contained "software product line and feature model", there were studies that contained feature modeling solutions at the architecture level which was clearly out of scope, (see e.g. (Zhu, et al., 2006) for an example). In Step 3, 359 studies were excluded leaving 270 studies in total. In Step 4, 208 studies out of 270 were excluded after reading the abstracts leaving 62 studies. The reason for excluding the 208 studies was that their focus, or main focus, was not domain analysis activities for software product lines. However, it was found that abstracts were of variable quality; some abstracts were missing, poor, and/or misleading, and several gave little indication of what was in the full article. In particular, it was not always obvious whether an experience report indeed included a domain analysis solution. If it was unclear from the title, abstract, and keywords whether a study conformed to the screening criteria, it was included giving it the benefit of doubt.

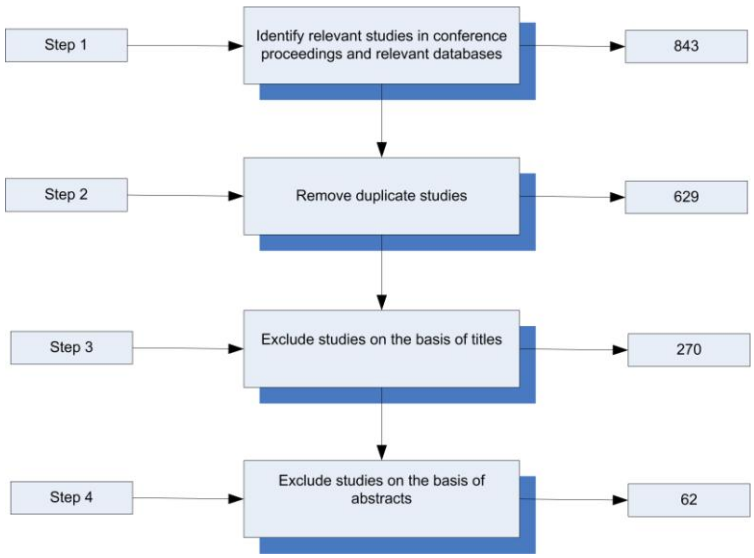

Figure 2 – Steps of Phase 2 of the search strategy

The inclusion and exclusion criteria were pilot-tested by the authors on a random sample of 15 studies. An agreement on inclusion and exclusion was achieved on 12 studies. The conflict on the remaining 3 papers was resolved after a discussion session and the inclusion/exclusion rules were refined. After the pilot, the primary author screened the remaining studies and marked them as included/excluded based on the approach described.

Phase 3: In order to ensure that no relevant studies were left out, three prominent researchers in the field of domain analysis and software product lines were consulted. As a result, three more studies were added.

This led to a selection of 89 studies in total relevant for this systematic review: 24 studies from Phase 1 + 62 studies from Phase 2 + 3 studies from Phase 3.

3.3.2 Quality Assessment and Data Extraction Procedure

The aim of the systematic review was to assess levels of empirical evidence and thus it did not impose any restriction in terms of any specific research method or experimental design, therefore the study quality assessment covered both quantitative and qualitative studies. The study quality assessment was primarily included in the inclusion criteria and scoping of the review, i.e. only studies that present any type of evidence or evaluation related to domain analysis for software product lines/families were included in the study. Moreover, the study quality assessment was used as a means to guide the interpretation of the findings.

Based on the research questions (see Section 3.1), a set of data extraction categories were identified with the help of guidelines from (Creswell, 2003) (Kitchenham, et al., 2002). Further, the categories were identified using the Goal Question Metric approach (GQM) (Basili, et al., 1994) during several brain storming sessions to ensure that categories identified address the aspects required to answer the research questions. Table 2 contains the definitions of the data extraction categories. There are only two categories for the research type as the purpose of the review is not to classify studies (Wieringa, et al., 2005) but rather find out how many and to what extent the proposed solutions are empirically applied and/or validated.

Table 2 – Definitions of the data extraction categories.

Research Type	
New Solution	Is it a new solution for domain analysis? This includes only scoping and/or modeling parts of domain analysis solution e.g. if a new method of feature modeling has been presented it is considered as a new solution.
Experience report	Is the paper an experience report describing the introduction of product lines in a company?
Empirical Basis	
Empirical	If it is "New Solution", is it developed based on empirically identified industry needs?
Non Empirical	It is a new research idea.
Basis Reported as	If a study has empirical basis then the empirical basis reported can be categorized into 1. Statements only: the authors have written statements claiming that the need for the proposed solution has been identified in industry. 2. Participation knowledge: the authors are either practitioners in industry or participate in industry work and have identified the need for the proposed solution through participation. 3. Interviews: the authors have conducted interviews with experts in industry to identify/confirm a need and have shown that the need for the proposed solution has been identified through those interviews. 4. Process assessment: the authors have undertaken some formal process assessment e.g. using CMMI, IDEAL, REPEAT etc. and identified the need for the proposed solution.
Application/Validation	
Empirical	It is applied/validated in laboratory setting or industry.
Non Empirical	It is not applied/validated in laboratory setting or industry.
Application/Validation Method	If a study contains empirical application/validation, what method of application/validation was used? Case study, survey, interviews, experiment, observations, other, as stated by the authors. We did not differentiate between research type and research context is because of the fact that almost all of the papers included in the review had industry as context, thus we only differentiated between research types. If it is mentioned in the paper that it has been applied/validated in industry but no description of application/validation method used is given then the method is classified as "Mentioned Industry Use".
Application/Validation Design Explained	If a study contains empirical application/validation, the level of explanation of the design/execution of the application/validation method used is categorized into: 1. Statements: authors stated that they have applied/validated the solution in industry but no summary/details as how this was done. 2. Application/Validation summary: summary of the method without details e.g. no research questions/hypothesis, context of study, sampling of population, study execution, validity threats and so on. 3. Application/Validation in detail: a detailed explanation of the application/validation method including research questions/hypothesis, context of study, sampling of population, study execution, validity threats and so on.
Application/Validation Results Explained	If a study contains empirical application/validation, the level of explanation of the application/validation results is categorized into: 1. Nothing: no information is stated as to the results of the application/validation in the paper. 2. Statements only: the authors have written statements about the results e.g. '*by applying the proposed solution, time to market decreased by 15%'*. This is a statement without any results or clarification of how the results were obtained. 3. Qualitative results: for example expert opinions, e.g. 4 experts were interviewed and they foresee that application of the proposed solution would result in 15% decrease in time to market. 4. Quantitative results: collected metrics and measurements are presented. 5. Qual + Quant: when a combination of qualitative and quantitative results are presented.
Driver of Validation	If a study contains empirical application/validation, who was driving the validation of the solution in industry, was it a researcher or a practitioner? Answers can be: researcher, practitioner.
Replication Study	Is it a replication study? The answer could be Yes, No, Not clear, N/A (not applicable).
Builds on Paper(s)	Does the current paper build on future work of some previous paper published at SPLC or uses and enhances any "New Solution" presented previously at SPLC? This does not include a paper that has been referenced in "Introduction" and/or "Related work" section. The answer could be Yes or No. If the answer to the question is yes, mention the paper it is related to.

Usability & Usefulness	
Usable	Yes/No
Usability Reported as	If a study reports usability of proposed solution, the level of usability reported can be categorized into 1. Statements: the authors have written statements claiming usability e.g. "*a recent BigLever customer was able to convert their existing one-of-a-kind product into a GEARS production line with three custom product instances in less than one day*" ((Krueger, 2002)). 2. Qualitative data: as expert opinions e.g. 4 experts were interviewed and based on their judgment they stated the solution can be introduced in 2 days and can be applied to products with 50-500 requirements. 3. Quantitative data: e.g. "*Document examination indicated that the team understood and was able to apply all notations used after only the four hour introduction to the approach, even though they had no earlier experience of feature modeling"* (Eriksson, et al., 2005). 4. Qual + Quant: qualitative and quantitative data proving scalability of introduction and/or scalability of use e.g. **Quantitative data for scalability of introduction*:***"*Document examination indicated that the team understood and was able to apply all notations used after only the four hour introduction to the approach, even though they had no earlier experience of feature modeling"* (Eriksson, et al., 2005). **Qualitative data for scalability of use:** "*Experts could not identify any scalability problems with the approach"* (Eriksson, et al., 2005).
Useful	Yes/No
Usefulness Reported as	If a study reports usefulness of proposed solution, the level of usefulness reported can be categorized into 1. Statements: the authors have written statements claiming usability e.g. "*It is an effective product line validation model*" (Mannion and Camara, 2003). 2. Qualitative data: as expert opinions e.g. 4 experts were interviewed and based on their judgment they stated the proposed solution will be effective than the existing method for requirements triage. 3. Quantitative data: proving effectiveness of the proposed solution e.g. data showing effective features representation and handling using proposed solution compared to existing solution. 4. Qual+Quant: qualitative and quantitative data proving effectiveness of proposed solution.
Future Work Mentioned	Has the paper made promises of future work in relation to the current research? The answer can be yes/no
Written by	Is the paper written by a practitioner? As stated above by practitioners it is meant the people working in industry. If the author is affiliated with industry research departments, the study is categorized as "written by practitioner". The answer can be yes/no/not clear.

The categorization of quality attributes (usability and usefulness) into quantitative and qualitative is not intended to indicate a preference or valuation of one over the other. Any empirical data (evidence) is judged on its own merits. For example, quantitative results obtained through a controlled experiment with students as subjects might not be as valuable as the expert opinion obtained in a case study with industry practitioners who actually applied a particular solution in industry. Moreover, context, background description and design also weigh in as the purpose is to categorize the reported empirical data to analyze the levels of usability and usefulness of a proposed solution. For example, a claim about the usability and/or usefulness of the presented solution without any description of context or how the claim can be substantiated is still considered as empirical evidence from the perspective of the study, but further analysis lets the reader weigh the value of the evidence.

In order to demonstrate the mapping between research questions and the design process, Table 3 shows the research questions and the corresponding data extraction categories.

Table 3 – Mapping between research questions with data extraction categories.

Research Questions	Data Extraction Categories
RQ1. Are proposed solutions based on needs identified in industry?	• Research type • Practicality – Empirical Basis • Builds on Paper(s)
RQ2. Are proposed solutions applied and/or validated in a laboratory setting or in industry?	• Research type • Practicality – Application/Validation a. Application/Validation Method b. Application/Validation Design Explained c. Application/Validation Results Explained d. Driver of Application/Validation

	e. Replication Study • Builds on Paper(s) • Future Work Mentioned • Written by
RQ3. Are the proposed solutions usable? RQ4. Are the proposed solutions usable?	• Usability & Usefulness a. Usability b. Usability Reported as c. Usefulness d. Usefulness Reported as

Similar to the inclusion/exclusion process, the data extraction process was tested by the authors using a sample of 10 included studies. An initial agreement on the data extraction was achieved for the data extracted from 9 studies which is quite high. For the remaining study, the consensus was reached in a discussion session. This was done to ensure that there was a common understanding of the categories defined and the classification was agreed upon by two researchers avoiding the potential bias and error source of having only one researcher performing the categorization.

3.4 Validity Evaluation

This section presents the different validity threats related to the review and how they were addressed prior to the study to minimize the likelihood of their realization and impact.

3.4.1 Conclusion Validity

Threats to conclusion validity are related with issues that affect the ability to draw the correct conclusions from the study (Wohlin, et al., 2000). From the review perspective, a potential conclusion validity threat is the reliability of the data extraction categories. To minimize this threat, GQM was used in several brain-storming sessions to extract the research questions and based on the research questions, measures (in this case are data extraction categories) were identified (see Section 3.2.3). In addition, the results presented in the review are not categorical. Any evidence, or claim made by authors are given the benefit of the doubt and counted as evidence. However, the claims are broken down and analyzed, and the value can be judged by the reader as every analysis and analysis step is transparently shown in the paper.

3.4.2 Construct Validity

Construct validity concerns generalizing the results of the study to the concept or theory behind the study (Wohlin, et al., 2000). From the review's perspective, a construct validity threat could be biased judgment. In this study the decision of which studies to include or exclude and how to categorize the studies could be biased and thus pose a threat. To minimize this threat both the processes of inclusion/exclusion and data extraction and coding were piloted prior to the study (see Section 3.3.1 and 3.3.2).

3.4.3 External Validity

The key idea with a systematic review is to capture as much as possible of the available literature to avoid all sorts of bias. The main challenge with a systematic review is the reliability. The reliability has been addressed as far as possible by involving two researchers, and by having a protocol which was piloted and hence evaluated. If the study is replicated by another set of researchers, it is possible that some studies that were removed in this review will be included and other studies would be excluded. However, it is highly unlikely that these random differences based on personal judgments would change the general results. It may change the actual numbers somewhat, but it is not likely that it would change the overall results as they are dominantly skewed towards one end of the spectrum (see Section 4). Thus, it is concluded that in general we believe that the external validity of the study is high given the use of a very systematic procedure, consultation with researchers in the field and involvement and discussion between the two researchers.

4. RESULTS AND ANALYSIS

This section presents a summary of the results of the inclusion/exclusion procedure (see Section **Error! Reference source not found.**) as well as the results of data extraction from the included studies (see Section 4.1).

4.1 Included Studies Overview

Summarizing the data extracted from the included studies, 48 studies[1] out of 89 studies (both "new solution" and "experience report" types) have some form of empirical basis, all 89 studies contain some form of application/validation, and out of these 64[2] are written by researchers. The remaining 25 studies[3] are written by practitioners. None of the 89 studies is a replication study. In total 36 studies[4] out of 89 have reported on some sort of usability, and 87[5] studies out of 89 have claimed usefulness in some form.

4.2 ANALYSIS

In this section the data extracted is analyzed with respect to the research questions posed in Table 1.

4.2.1 RQ1 (Are proposed solutions based on needs identified from Industry?)

Almost half of the studies are based in some sense on the needs identified in industry (see Section 4.1). However, a deeper analysis of the empirical basis reported can be seen in Table 4 which shows that a majority of the studies have mentioned identified needs as “Statements only” (42% studies[6]), or as “Participation knowledge” (50% studies[7]). Only 2% studies[8] have mentioned interviewing experts to identify needs, and only 6% studies[9] have stated that some form of process assessment was used to identify the need for the proposed solution.

These results make it hard to judge the credibility of the empirical basis of the solutions proposed due to the absence of presentation of e.g. process assessment and/or experts’ opinions through e.g. interviews. In addition, due to the almost total lack of how the practitioners knew about the problems/needs that constitute the basis for the solutions proposed, it is impossible to draw any conclusions. In the few cases where process assessment or interviews were conducted no details such as selection criteria, method used or number of interviews, and so on are explained.

Table 4 – Included studies, “Basis” categorization

Basis Reported as	Number of Studies
Statements only	20
Participation knowledge	24
Interviews	1
Process assessment	3
Total	**48**

Moreover, although a majority of the studies claims empirical basis, very few are based on future work described by previously published studies, or extend previously published solutions[10]. This may indicate that in the absence of expert interviews or proper process assessments, the needs identified may not be representative of the current problem or valid for other companies in similar situations.

The answer to RQ1 is that a majority of the proposed solutions are based on needs identified in industry, however, the actual method used and the validity of the results is impossible to ascertain as very little information is given.

4.3 RQ2 (Are proposed solutions applied and/or validated in a laboratory setting or in industry?)

An analysis of the studies, claiming application/validation (see Figure 3) reveals that for the years 1998 to 2007, 33% studies[11] have used case study as an application/validation method. In 24% of the studies[12] industry use was only stated and 36% studies[13] have demonstrated application/validation through simplified examples (top left quadrant). This means that 60% of the applied/validated studies have either only mentioned industry use without any details reported, or have used simplified examples to demonstrate practicality of a proposed solution. The remaining 40% have described some details about application/validation. This makes it harder to judge the scalability of introduction and scalability of use of the proposed solutions. From Figure 3, it is evident that only 5% studies[14] have used workshops, pilots and prototyping. It is also interesting to note that for the years 1998 to 2007 only 2% studies[15] have used experimentation as a validation method.

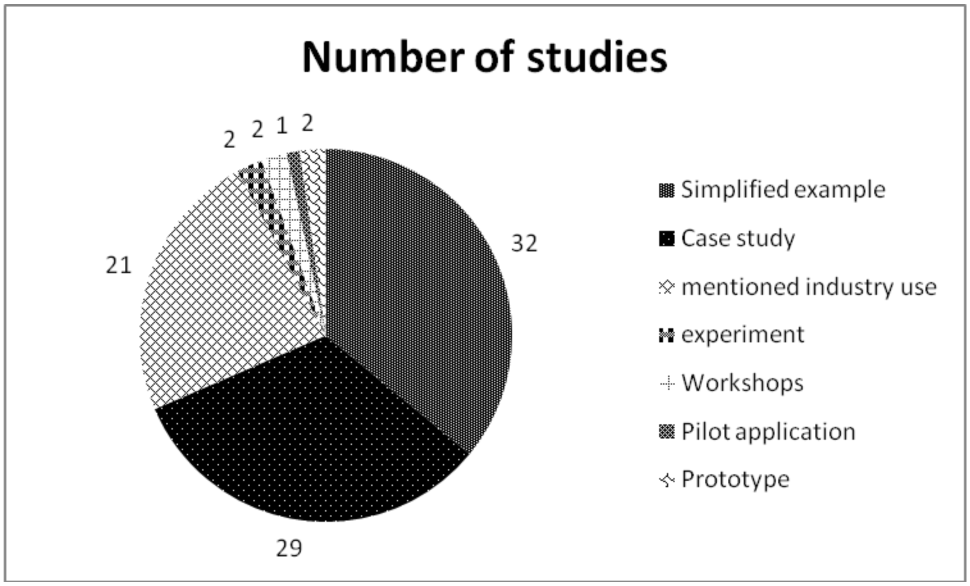

Figure 3 – Number of studies categorized according to the application/validation method

One of the reasons for these numbers could be that it is difficult to do experimentation for new solutions for product lines due to complexity of the area and difficulty in covering the entire scope in a controlled experiment. However, empirical studies are the building blocks essential for collecting evidence and to determine what situations are best for using a particular solution (Pfleeger, 1999). The current situation means there is a lack of quantitative and/or qualitative data that a new solution is better than an already existing one, or what the impact of implementing the new solution might be. This makes it impossible to gauge efficiency or effectiveness of proposed solutions either alone or in relation to better alternative investment (BAI).

Table 5 – Included studies, "Application/Validation Design Explained" categorized.

Application/Validation Design Explained	Number of Studies
Statements only	14
Application/Validation summary	45
Application/Validation in detail	30
Total	**89**

Moving on from the analysis of application/validation methods used to the analysis of the application/validation design details, Table 5 shows the categorization of the application/validation design explanation given in the included studies claiming some form of application/validation. From Table 5, it is possible to see that 84% studies either provide application/validation summary (50% studies[16]) or explain application/validation in detail (33% studies[17]). This seems to be a positive outcome that most of the studies have explained application/validation in detail. However, after analyzing the level of application/validation results, it is found that a majority of the studies either say nothing about the application/validation results (11%) or have only statements about the results (68% studies) (see first two rows of Table 6). Only 6% studies[18] provide qualitative results as experts' opinion, 8% studies[19] provide quantitative results and only 3% studies[20] provide both qualitative and quantitative results.

Table 6 – Included studies "Application /Validation Results" categorized.

Application/Validation Results Explained	Number of Studies
Nothing	10
Statements only	61
Results as experts opinion	6
Quantitative data	7
Qual +Quant	3
Total	**89**

Thus, Table 6 reveals that out of 89 included studies with the claim of empirical evidence, 80% studies lack qualitative or quantitative results of application/validation. The absence of strong application/validation results may be one of the reasons that few studies have used previously proposed solutions (see Section 4.2.1).

Majority of the "Experience report" studies state the results of the experiences as lessons learned without any indication how these lessons were collected. The lack of description in relation to the experiences, for example if interviews were used, if there were any quantitative measures and so on, makes it difficult to judge validity. This also makes it hard for other practitioners to gauge the context and relevance of the experiences reported.

Figure 4 presents another aspect to answer this question and that is to see how many solutions from each year are based on solutions presented previous years. Figure 4 shows that many new solutions have been presented over the years, but very few actually have been used as a basis for further development or adoption, piloting or test in industry. For example, by the year 2003 a total of 28 new solutions had been proposed but only 5 studies reported the use of any of the previously proposed solutions (in industry or as a basis for refinement of a solution). By 2007 the number of "New Solution" studies had reached 73, and only 12 studies were based on previously proposed solutions or reported experience based on the use of previously proposed solutions. This can indicate that the proposed solutions are not applicable in industry or that due to missing application/validation results the solutions are not applied by practitioners and not used by researchers. This problem has been indicated by others as well e.g. in (Kircher, et al., 2006). This can imply that a focus on validation and proper reporting should be premiered over the continuous presentation of new solutions. Another possibility is that the proposed solutions do not solve the challenges in industry, which in turn implies that there is a need to understand the challenges. Another possible conclusion could be that industry practitioners are not up to date with the new solutions proposed, thus the solutions go unused. None of the studies presented from the year 1998 to 2007 were replicated studies.

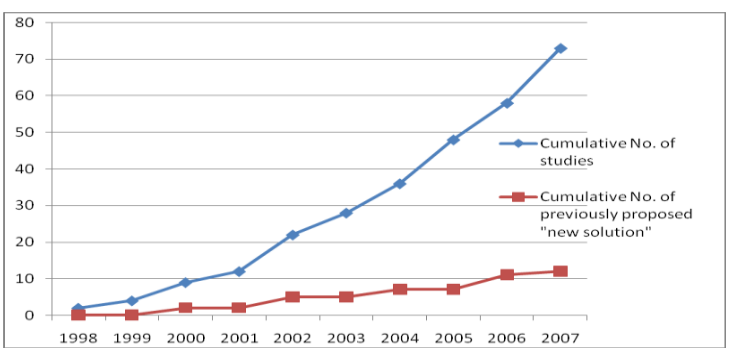

Figure 4 – Cum No. of studies VS Cum No. of previously proposed "new solution" studies

Summarizing all the aspects analyzed above, the answer to RQ2 is that although the studies from the years 1998 to 2007 have reported some form of application/validation, the absence of detailed results or replicated studies make it difficult to evaluate the potential of the proposed solutions. This is further compounded by the fact that the validity of the reported experiences reports is also very hard to judge.

4.4 Answering RQ3 (Are the proposed solutions usable?) and RQ4 (Are the proposed solutions useful?)

In 36 studies usability was mentioned as a part of the proposed solutions. However, looking at Table 7 it is possible to see that 80% of the studies[21] only have statements claiming usability. An example of this can be illustrated by the following statement*: "A minor problem occurs as the table can grow and become unwieldy for large application areas, but this can be addressed by segmenting the table appropriately*"(DeBaud and Schmid, 1999). In 8% of the studies[22] qualitative evidence of usability as expert opinion was presented, for example, "*After finishing the project, the project manager and developers agreed that the proposed domain requirements development approach was very helpful for identifying and specifying application requirements, resulting in reducing the overall development effort*" (Moon, et al., 2005). In 8% of the studies[23] gave quantitative evidence of usability and only 2% of the studies[24] gave both qualitative and quantitative evidence of usability e.g. "…*indicated that the team understood and was able to apply all notations used after only the four hour introduction to the approach, even though they had no earlier experience of feature modeling*", results of "*questionnaire indicated that the product line analysis team gained a better understanding of the domain during the modeling activity*" (Eriksson, et al., 2005).

Table 7 – "Usability" reported in included studies.

Usability	Number of studies
Statements only	29
Results as experts opinion	3
Quantitative results	3
Qual +Quant	1
Total	**36**

Clearly with the usability statements as exemplified above, it is difficult to judge the usability of a proposed solution. A clear majority of the studies do not include either qualitative or quantitative data about solution scalability of introduction or scalability of use, making it harder for practitioners to evaluate usability. It is important to understand that the intention in this review is not to criticize studies but to highlight the absence of qualitative and quantitative evidence in relation to usability, which might be a barrier for the industrial adoption of the proposed solutions.

Table 8 - "Usefulness" reported in included studies.

Usefulness	
Statements only	70
Results as experts opinion	9
Quantitative results	6
Qual +Quant	2
Total	**87**

Positive results regarding usefulness were reported by 87 of the included studies, which seem to be very good, but a deeper analysis as can be seen in Table 8 shows that 81% of the studies[25] claim usefulness as statements. For example, "*The ILP modeling approach presented in the former section was tested in a Stago project with satisfying results*" (Djebbi and Salinesi, 2007). There are only 2% of the studies[26] that provides qualitative and quantitative data about the usefulness of the proposed solution.

If the percentages and the categorization of reported application/validation results are kept in mind, it seems logical that since the application/validation results were mostly statements (see Table 6), usability and usefulness evidence would naturally also be statements due to the absence of qualitative and/or quantitative results for the application/validation of a solution. This also results in difficulty to find any qualitative or quantitative evidence of usability or usefulness, even in the form of statements and claims, made by the authors of respective studies.

The answer to RQ3 and RQ4 is that although there are statements regarding usability and usefulness in the studies published for the years 1998 to 2007, lack of qualitative and quantitative data of any sort makes it difficult to evaluate how usable and useful the proposed solutions/experiences are. In industry, time and resources are scarce. If a practitioner cannot clearly determine the time and resources required to implement a solution against the usefulness of the solution in comparison to available better alternative investments, it is very unlikely that the solution will be adopted based only on statements made by the creators of the solution. Similarly, if authors do not show scalability of use of a particular solution indication the ability to tackle industry grade problems, practitioners would probably not take the risk of implementing a solution, as it falls short on reporting even rudimentary evidence on efficiency.

5. CONCLUSION

This paper presents the systematic review of the modeling and scoping activities involved in domain analysis for software product lines from the year 1998 until 2007. With a three phase search strategy 89 studies were selected that either proposed new solutions of domain analysis, or reported of experiences in using such solutions. In order to *analyze the practical application and validation of proposed domain analysis solutions in industry and to gauge their practical usability and usefulness*, four research questions were specified (see Section 3.1.). Based on the goal and corresponding research questions to achieve that goal, a data extraction procedure was defined (see Section 3.3.2.). Data was then extracted using a defined procedure covering the basis of a study, practicality, usability and usefulness, future work and information about the authors.

The major findings of the review can be summarized as follows:

1) Many domain analysis solutions have been presented over the years and a majority of the studies address needs identified in industry, but they fall short on the strength of the approach used to identify the need for a solution. Most studies only claim that they based the solution on a need identified in industry or state that through participation knowledge the need for the proposed solution was identified. Such claims and statements may be valid, but they raise validity questions both from a research perspective and an industrial adoption perspective. Without interviewing experts in industry or performing some form of process assessment, it is hard to triangulate the need identified thus raising the issue that the need may not be representative of the current situation. As a result, this poses questions about the internal and external validity of the needs identified, and this is passed on to the corresponding solutions proposed.
2) Many studies claim that they have applied/validated the proposed solutions in industry; however, a deeper analysis reveals that a majority of the claims are merely statements (80%), and qualitative and quantitative evidence supporting these claims is generally missing. Claims and statements may be valid, but in the absence of clear qualitative evidence as experts' opinions and/or quantitative data about the benefits of the proposed solution, it is hard to evaluate the potential of these solutions for industry adoption.
3) Many studies claim usability and usefulness of the proposed solutions in some form, however a deeper analysis reveals that majority of the claims are also merely statements about usability (80%) and usefulness (81%). As mentioned previously such claims may be valid, but they raise validity questions from both a research and industrial adoption perspective. Without experts' opinions and/or quantitative data supporting the usability and usefulness claims, it is difficult to evaluate the validity of the claims, and similarly it is difficult for the practitioners to evaluate the usability and usefulness of a proposed solution for application in industry.

The overall goal of this review was not to expect or demand perfect evidence of usability and usefulness following perfect and extensive data collection in industry. However, many studies over the years have shown that it is possible to validate solution suggestions in any number of ways. Controlled experiments could be used in academia, even if the use of students is debated. Static (preliminary) validation can be performed in industry through workshop, interviews, or surveys. Dynamic validation (e.g. pilots) can be performed collecting metrics and qualitative data though interviews with practitioners. The data collected is not complete, but vastly better than no data at all.

In addition to doing validation (e.g. in industry), the way in which the validation is planned and reported is also crucial. The studies reviewed are full of statements, claiming usability and usefulness. The good thing is that this indicates that our interest in these two concepts in this systematic review is relevant, i.e. usability and usefulness of solutions are important and this is confirmed by the authors themselves. However, even if statements are common, very little evidence is presented, both in terms of absence of data, but also absence of design for the studies presented. The only seemingly complete validation is when there is no real validation, e.g. in case of presenting simplified examples. The use of simplified examples is not without merit, e.g. it can be used to explain and exemplify the use of a method initially, but the use of a simplified example is not the same as validation, even if the example is based on something relevant for industry. One might even go so far as to expect an evolution, that is, a new solution proposed is exemplified and explained through the use of simplified and scaled down examples in initial publications, then validation is performed, scaling up the tests of the solution.

The presence of empirical evidence of any sort with at least some intent to explain the overall design and execution of a validation (e.g. a pilot test in industry) could be very beneficial for both researchers and industry practitioners. From an academic point of view the possibility to learn and extend on presented research is crucial for progress. In addition, one of the foundations of research is the possibility to replicate studies. None of the studies included was a replicated study.

From a practitioner point of view, a design and illustration of how conclusions about usability and usefulness are made can vastly improve the relevance of any paper. The total absence of data or evidence is problematic from two perspectives. First, can the results be trusted? Second, even if the authors are given the benefit of the doubt, is the proposed solution relevant for all cases? If not, what cases?

There can be several explanations for the results of this systematic review. One could be that the included conference and journals attract a certain type of studies that do not focus on empirical results. Another explanation could be that in case of conferences a ten page limit presents problems for presenting empirical results, even if there are many studies who manage (some examples from SPLC conference are (Eriksson, et al., 2005) and (Jepsen, et al., 2007, Lee, et al., 2000)). Yet another explanation could be that industry validation is hard to achieve. The question is, should we accept these explanations, or should we strive for improving state-of-the-art reporting?

Summarizing the contribution of this systematic review we have two main perspectives. For industry practitioners looking to adopt a domain analysis solution, the results of the study can be used as an indication of maturity as well as to estimate potential risk of a certain solution before considering its application. From an academic point of view researchers planning studies and evaluation of a solution can use this study as inspiration for study design as the evaluation criteria of the review presented in this paper could be seen as a checklist to ascertain usability and usefulness.

6. REFERENCES

P. Clements and L. Northop, 2001, Software Product Lines: Practices and Patterns, Addisson-Wesley, 0-201-70332-7

S. Deelstra, M. Sinnema and J. Bosch, 2004, Experiences in Software Product Families: Problems and Issues During Product Derivation, Proceedings of Third International Conference, SPLC2004, Boston, MA, USA, Springer

M. Kim, S. Park, V. Sugumaran and H. Yang, 2007, Managing requirements conflicts in software product lines: A goal and scenario based approach, Data and Knowledge Engineering, 61, 3, 417-432

A. H. William, W. K. Charles and G. M. Joseph, 2006, Incremental return on incremental investment: Engenio's transition to software product line practice, Companion to the 21st ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications, Portland, Oregon, USA, ACM

J. C. Dager, 2000, Cummins's experience in developing a software product line architecture for real-time embedded diesel engine controls, Proceedings of the first conference on Software product lines : experience and research directions: experience and research directions, Denver, Colorado, United States, Kluwer Academic Publishers

K. Pohl, G. Boeckle and F. van der Linden, 2005, Software Product Line Engineering, Springer, 10 3-540-24372-0 Springer Berlin Heidelberg New York

M. Staples and D. Hill, 2004, Experiences adopting software product line development without a product line architecture,

G. Bockle, 2000, Model-based requirements engineering for product lines, Proceedings of the first conference on Software product lines : experience and research directions: experience and research directions, Denver, Colorado, United States, Kluwer Academic Publishers

P. America, S. Thiel, S. Ferber and M. Mergel, 2001, Introduction to Domain Analysis,

D. I. K. Sjoeberg, J. E. Hannay, O. Hansen, V. B. Kampenes, A. Karahasanovic, N. K. Liborg and A. C. Rekdal, 2005, A survey of controlled experiments in software engineering, Software Engineering, IEEE Transactions on, 31, 9, 733-753

V. Ramesh, L. Glass Robert and I. Vessey, 2004, Research in computer science: an empirical study, The Journal of Systems and Software, 70, 1-2, 165-176

R. L. Glass, I. Vessey and V. Ramesh, 2002, Research in software engineering: An analysis of the literature, Information and Software Technology, 44, 8, 491-506

D. E. Perry, A. A. Porter and L. G. Votta, 2000, Empirical studies of software engineering: a roadmap, Proceedings of the Conference on The Future of Software Engineering, Limerick, Ireland, ACM

E. Mendes, 2005, A systematic review of Web engineering research, International Symposium on Empirical Software Engineering, 2005.,

R. Wieringa, N. Maiden, N. Mead and C. Rolland, 2005, Requirements engineering paper classification and evaluation criteria: a proposal and a discussion, Requir. Eng., 11, 1, 102-107

T. Gorschek, C. Wohlin, P. Garre and S. Larsson, 2006, A model for technology transfer in practice, IEEE Software, 23, 6, 88-95

B. A. Kitchenham, 2007, Guidelines for performing Systematic Literature Reviews in Software Engineering, Technical Report EBSE-2007-01,

C. Zhu, Y. Lee, W. Zhao and Z. Jingzhou, 2006, A feature oriented approach to mapping from domain requirements to product line architecture, Proceedings of the 2006 International Conference on Software Engineering Research and Practice and Conference on Programming Languages and Compilers SERP'06, USA, CSREA Press

J. W. Creswell, 2003, Research Design: Qualitative, Quantitative, and Mixed Methods Approaches, Sage Publications,
B. A. Kitchenham, S. L. Pfleeger, L. M. Pickard, P. W. Jones, D. C. Hoaglin, K. El Emam and J. Rosenberg, 2002, Preliminary guidelines for empirical research in software engineering, IEEE Transactions on Software Engineering, 28, 8, 721-34
V. Basili, G. Caldiera and H. D. Rombach, 1994, The Goal Question Metric Approach, John Wiley & Sons,
C. W. Krueger, 2002, Easing the Transition to Software Mass Customization, Revised Papers from the 4th International Workshop on Software Product-Family Engineering, Springer-Verlag
M. Eriksson, J. Börstler and K. Borg 2005, The PLUSS Approach - Domain Modeling with Features, Use Cases and Use Case Realizations, Ninth International Conference on Software Product Lines, Rennes, France,
M. Mannion and J. Camara, 2003, Theorem proving for product line model verification, 5th International Workshop Software Product - Family Engineering., Siena, Italy,
C. Wohlin, P. Runeson, M. Höst, M. C. Ohlson, B. Regnell and A. Wesslén, 2000, Experimentation in Software Engineering: An Introduction, Kluwer Academic, 0792386825 (acid-free paper)
S. L. Pfleeger, 1999, Albert Einstein and empirical software engineering, Computer, 32, 10, 32-38
M. Kircher, C. Schwanninger and I. Groher, 2006, Transitioning to a Software Product Family Approach - Challenges and Best Practices, Proceedings of the 10th International on Software Product Line Conference, IEEE Computer Society
J. M. DeBaud and K. Schmid, 1999, A systematic approach to derive the scope of software product lines, Proceedings of the 21st international conference on Software …,
D. Jean-Marc and S. Klaus, 1999, A systematic approach to derive the scope of software product lines, Proceedings of the 21st international conference on Software engineering, Los Angeles, California, United States, ACM
M. Moon, K. Yeom and H. S. Chae, 2005, An approach to developing domain requirements as a core asset based on commonality and variability analysis in a product line, IEEE Transactions on Software Engineering, 31, 7, 551-569
O. Djebbi and C. Salinesi, 2007, RED-PL, a method for deriving product requirements from a product line requirements model, Springer Verlag, Heidelberg, D-69121, Germany
H. P. Jepsen, J. G. Dall and D. Beuche, 2007, Minimally Invasive Migration to Software Product Lines, Proceedings of the 11th International Software Product Line Conference, IEEE Computer Society
K. Lee, K. C. Kang, E. Koh, W. Chae, B. Kim and B. W. Choi, 2000, Domain-oriented engineering of elevator control software: a product line practice, Proceedings of the first conference on Software product lines : experience and research directions: experience and research directions, Denver, Colorado, United States, Kluwer Academic Publishers

Appendix A – Search Strings

Search Strings	Databases
(((("product famil*" OR "product line*") AND ({requirements} OR {requirements engineering*} OR {conceptual model*} OR {requirements model*} OR {commonality and variability model*} OR {domain model*} OR {feature model*} OR {scenario model*} OR {commonality analysis} OR {variability analysis} OR {domain eval*} OR {domain scop*} OR "asset scop*")) AND ((((("product famil*" OR "product line*") AND ({requirements} OR {requirements engineering*} OR {conceptual model*} OR {requirements model*} OR {commonality and variability model*} OR {domain model*} OR {feature model*} OR {scenario model*} OR {commonality analysis} OR {variability analysis} OR {domain eval*} OR {domain scop*} OR "asset scop*")) AND (empiric* OR experience* OR "lesson learned" OR "lesson learnt" OR "lessons learned" OR evaluat* OR validat* OR experiment* OR stud* OR case* OR example* OR survey OR analys* OR investig* OR demonstrate* OR industr*) WN KY) AND (1969-2007 WN YR) AND (English) WN LA)) WN KY) AND (1969-2007 WN YR) AND (English) WN LA)	Inspec and Compendex via Engineering Village2
(((Abstract:product and Abstract:line) OR (Abstract:product and Abstract:famil*)) AND ((Abstract:requirements and Abstract: Model*) OR (Abstract:requirements and Abstract: engineer*) OR (Abstract:requirements) OR (Abstract:conceptual and Abstract:Model*) OR (Abstract:feature and Abstract: Model*) OR (Abstract:commonality and Abstract:analysis) OR (Abstract:variability and Abstract:analysis) OR (Abstract:domain and Abstract:scop*) OR (Abstract:domain and Abstract:eval*) OR (Abstract:Asset and Abstract:scop*)) AND ((Abstract:case and Abstract:stud*) OR (Abstract:empiric*) OR (Abstract:experienc*) OR (Abstract:lessons and Abstract:learn*) OR (Abstract:evaluate*) OR (Abstract:validate*) OR (Abstract:experiment*) OR (Abstract:example*) OR (Abstract:survey*) OR (Abstract:analy*) OR (Abstract:investigat*) OR (Abstract:demonstrat*) OR (Abstract:industr*)))	ACM
((((product line)<in>ab) <or> ((product famil*)<in>ab)) <and> (((requirements model*)<in>ab) <or> ((requirements engineer*)<in>ab) <or> ((requirements)<in>ab) <or> ((conceptual model*)<in>ab) <or> ((feature model*)<in>ab) <or> ((commonality analysis)<in>ab) <or> ((variability analysis)<in>ab) <or> ((domain sco*)<in>ab) <or> ((domain eval*)<in>ab) <or> ((asset sco*)<in>ab)) <and> (((experienc*)<in>ab) <or> ((empiric*)<in>ab)<or> ((lessons learn**)<in>ab)<or> ((evaluat*)<in>ab)<or> ((validat*)<in>ab)<or> ((expeiment*)<in>ab)<or> ((case study)<in>ab)<or> ((survey*)<in>ab)<or> ((analy*)<in>ab)<or> ((investigat*)<in>ab)<or> ((demonstrat*)<in>ab)<or> ((industr*)<in>ab)))	IEEEXplore
TS=("product line" OR "product famil*" AND (TS=("requirements" OR "requirements engineering" OR "requirements model*" OR "feature model*" OR "commonality analysis" OR "variability analysis" OR "domain scop*" OR "domain eval*" OR "asset scop*")) AND (TS=("case stud*" OR "empiric*" OR "experien" OR "Lessons learn*" OR "evaluat*" or "validat*" OR "experiment" OR "exampl" OR "survey" OR "Analy*" OR "investigat*" OR "validat*" OR "industri*"))) AND Language=(English) AND Document Type=(Article)	ISI Web of Sceince

Appendix B – Selected Studies

Study Id	Study Name
1	S. Jarzabek, B. Yang and S. Yoeun, 2006, Addressing quality attributes in domain analysis for product lines, IEE Proceedings: Software, 153, 2, 61-73
2	J. M. Thompson and M. P. E. Heimdahl, 2003, Structuring product family requirements for n-dimensional and hierarchical product lines, Requirements Engineering, 8, 1, 42-54
3	P. Knauber, D. Muthig, K. Schmid and T. Widen, 2000, Applying product line concepts in small and medium-sized companies, Ieee Software, 17, 5,
4	L. Soon-Bok, K. Jin-Woo, S. Chee-Yang and B. Doo-Kwon, 2007, An approach to analyzing commonality and variability of features using ontology in a software product line engineering, 5th International Conference on Software Engineering Research, Management and Applications, Piscataway, NJ, USA, IEEE
5	C. Kun, Z. Wei, Z. Haiyan and M. Hong, 2005, An approach to constructing feature models based on requirements clustering, Proceedings of 13th IEEE International Conference on Requirements Engineering, Los Alamitos, CA, USA, IEEE Comput. Soc
6	L. Yuqin, Y. Chuanyao, Z. Chongxiang and Z. Wenyun, 2006, An approach to managing feature dependencies for product releasing in software product lines, 9th International Conference on Software Reuse, ICSR 2006. Proceedings (Lecture Notes in Computer Science Vol.4039), Berlin, Germany, Springer-Verlag
7	C. Kuloor and A. Eberlein, 2003, Aspect-oriented requirements engineering for software product lines, Proceedings 10th IEEE International Conference and Workshop on the Engineering of Computer-Based Systems, Los Alamitos, CA, USA, IEEE Comput. Soc
8	R. R. Lutz and F. Qian, 2005, Bi-directional safety analysis of product lines, Journal of Systems and Software, 78, 2, 111-27
9	W. Lam, 1998, A case-study of requirements reuse through product families, Annals of Software Engineering, 5, 253-77
10	M. Ramachandran and P. Allen, 2005, Commonality and variability analysis in industrial practice for product line improvement, Software Process Improvement and Practice, 10, 1, 31-40
11	K. Schmid, 2002, A comprehensive product line scoping approach and its validation, Proceedings of the 24th International Conference on Software Engineering. ICSE 2002, New York, NY, USA, ACM
12	D. Jean-Marc and S. Klaus, 1999, A systematic approach to derive the scope of software product lines, Proceedings of the 21st international conference on Software engineering, Los Angeles, California, United States, ACM
13	J. Bayer, O. Flege, P. Knauber, R. Laqua, D. Muthig, K. Schmid, T. Widen and J. M. DeBaud, 1999, PuLSE: A methodology to develop software product lines, SSR'99. Proceedings of the Fifth Symposium on Software Reusability. Bridging the Gap Between Research and Practice, New York, NY, USA, ACM
14	M. Sinnema, S. Deelstra, J. Nijhuis and J. Bosch, 2004, COVAMOF: a framework for modeling variability in software product families, Third International Conference on Software Product Lines, SPLC 2004. Proceedings (Lecture Notes in Comput. Sci. Vol.3154), Berlin, Germany, Springer-Verlag
15	D. Aubrey, 2006, Controlling the HMS Program through managing requirements, Proceedings of the IEEE International Conference on Requirements Engineering, Piscataway, NJ 08855-1331, United States, Institute of Electrical and Electronics Engineers Computer Society
16	K. Schmid and I. John, 2004, A customizable approach to full lifecycle variability management, Science of Computer Programming, 53, 3, 259-284
17	T. Eisenbarth, R. Koschke and D. Simon, 2001, Derivation of feature component maps by means of concept analysis, Fifth European Conference on Software Maintenance and Reengineering, 2001,
18	O. Djebbi, C. Salinesi and D. Diaz, 2007, Deriving product line requirements: The RED-PL guidance approach, Proceedings of Asia-Pacific Software Engineering Conference, APSEC, Los Alamitos, CA 90720-1314, United States, IEEE Computer Society
19	S. Deelstra, M. Sinnema, J. Nijhuis and J. Bosch, 2004, COSVAM: A technique for assessing software variability in software product families, IEEE International Conference on Software Maintenance, ICSM, Los Alamitos, CA 90720-1314, United States, IEEE Computer Society

20	A. Metzger, P. Heymans, K. Pohl, P. Y. Schobbens and G. Saval, 2007, Disambiguating the documentation of variability in software product lines: a separation of concerns, formalization and automated analysis, 2007 IEEE International Conference on Requirements Engineering, Piscataway, NJ, USA, IEEE
21	K. Minseong, Y. Hwasil and P. Sooyong, 2003, A domain analysis method for software product lines based on scenarios, goals and features, 10 Asia-Pacific Software Engineering Conference, Los Alamitos, CA, USA, IEEE Comput. Soc
22	I. John, D. Muthig, P. Sody and E. Tolzmann, 2002, Efficient and systematic software evolution through domain analysis, Proceedings. IEEE Joint International Conference on Requirements Engineering, 2002.
23	A. K. Thurimella and B. Bruegge, 2007, Evolution in product line requirements engineering: a rationale management approach, 2007 IEEE International Conference on Requirements Engineering, Piscataway, NJ, USA, IEEE
24	M. Breen, 2005, Experience of using a lightweight formal specification method for a commercial embedded system product line, Requirements Engineering, 10, 2, 161-72
25	H. Zuo, M. Mannion, D. Sellier and R. Foley, 2005, An extension of problem frame notation for software product lines, Proceedings of Asia-Pacific Software Engineering Conference, APSEC, Los Alamitos, CA 90720-1314, United States, IEEE Computer Society
26	B. Jongsu and K. Sungwon, 2007, A Method to Generate a Feature Model from a Business Process Model for Business Applications, 7th IEEE International Conference on Computer and Information Technology, 2007. CIT 2007.,
27	K. C. Kang, L. Jaejoon and P. Donohoe, 2002, Feature-oriented product line engineering, IEEE Software, 19, 4, 58-65
28	S. Thiel and A. Hein, 2002, Modeling and using product line variability in automative systems, IEEE Software, 19, 4, 66-72
29	J. Kim, M. Kim, H. Yang and S. Park, 2004, A method and tool support for variant requirements analysis: Goal and scenario based approach, Proceedings of Asia-Pacific Software Engineering Conference, APSEC, Los Alamitos, CA 90720-1314, United States, IEEE Computer Society
30	M. Moon, K. Yeom and H. S. Chae, 2005, An approach to developing domain requirements as a core asset based on commonality and variability analysis in a product line, IEEE Transactions on Software Engineering, 31, 7, 551-569
31	H. Zhang and S. Jarzabek, 2004, XVCL: A mechanism for handling variants in software product lines, Science of Computer Programming, 53, 3, 381-407
32	H. Zhang and S. Jarzabek, 2004, XVCL: A mechanism for handling variants in software product lines, Science of Computer Programming, 53, 3, 381-407
33	A. K. Jeong, 2006, Variability management with ACM (Adaptable Component Model) for insurance product line, Proceedings of the 2006 IEEE International Conference on Information Reuse and Integration, IRI-2006, Piscataway, NJ 08855-1331, United States, Institute of Electrical and Electronics Engineers Computer Society
34	P. Prasanna and R. R. Lutz, 2005, Tool-supported verification of product line requirements, Automated Software Engineering, 12, 4, 447-65
35	T. Nakanishi, M. Fujita, S. Yamazaki, N. Yamashita and S. Ashihara, 2007, Tailoring the Domain Engineering Process of the PLUS Method, 14th Asia-Pacific Software Engineering Conference, 2007. APSEC 2007.
36	Y. P. Shin and D. K. Soo, 2005, A systematic method for scoping core assets in product line engineering, Proceedings - Asia-Pacific Software Engineering Conference, APSEC, Los Alamitos, CA 90720-1314, United States, IEEE Computer Society
37	M. Ardis, N. Daley, D. Hoffman, H. Siy and D. Weiss, 2000, Software product lines: a case study, Software-Practice & Experience, 30, 7, 825-847
38	G. Spanoudakis, A. Zisman, E. Perez-Minana and P. Krause, 2004, Rule-based generation of requirements traceability relations, Journal of Systems and Software, 72, 2, 105-127
39	L. Yiyuan, Y. Jianwei, S. Dongcai, L. Ying and D. Jinxiang, 2007, Software product line oriented feature map, Proceedings of 7th International Conference of Computational Science-ICCS 2007, Part II (Lecture Notes in Computer Science Vol.4488), Berlin, Germany, Springer
40	M. Eriksson, J. Borstler, and K. Borg, "Software product line modeling made practical," Communications of the ACM, vol. 49, p. 1183265, 2006.

41	H. Gomaa and M. Saleh, "Software product line engineering and dynamic customization of a radio frequency management system," Piscataway, NJ 08855-1331, United States, 2006, pp. 345-352.
42	H. Gomaa and M. E. Shin, "Multiple-view meta-modeling of software product lines," Los Alamitos, CA, USA, 2002, pp. 238-46.
43	M. A. Laguna, B. Gonzalez-Baixauli, and J. M. Marques, "Seamless development of software product lines," New York, NY 10036-5701, United States, 2007, pp. 85-94.
44	P. Sooyong, K. Minseong, and V. Sugumaran, "A scenario, goal and feature-oriented domain analysis approach for developing software product lines," Industrial Management + Data Systems, vol. 104, pp. 296-308, 2004.
45	J. Liu, J. Dehlinger, and R. Lutz, "Safety analysis of software product lines using state-based modeling," Journal of Systems and Software, vol. 80, pp. 1879-1892, 2007.
46	R. R. Lutz, G. G. Helmer, M. M. Moseman, D. E. Statezni, and S. R. Tockey, "Safety analysis of requirements for a product family," in Requirements Engineering, 1998. Proceedings. 1998 Third International Conference on, 1998, pp. 24-31.
47	F. Loesch and E. Ploedereder, "Restructuring variability in software product lines using concept analysis of product configurations," Los Alamitos, CA, USA, 2007, p. 10 pp.
48	J. Kuusela and J. Savolainen, "Requirements engineering for product families," Los Alamitos, CA, USA, 2000, pp. 61-69.
49	O. Djebbi and C. Salinesi, "RED-PL, a method for deriving product requirements from a product line requirements model," Berlin, Germany, 2007, pp. 279-93.
50	P. C. Clements, L. G. Jones, L. M. Northrop, and J. D. McGregor, "Project management in a software product line organization," IEEE Software, vol. 22, pp. 54-62, 2005.
51	S. R. Faulk, "Product-Line Requirements Specification (PRS): An approach and case study," 2001, pp. 48-55.
52	K. Schmid, "The product line mapping approach to defining and structuring product portfolios," Los Alamitos, CA, USA, 2002, pp. 219-26.
53	K. Schmid, "The product line mapping approach to defining and structuring product portfolios," Los Alamitos, CA, USA, 2002, pp. 219-26.
54	M. Aoyama, "Persona-and-scenario based requirements engineering for software embedded in digital consumer products," in Requirements Engineering, 2005. Proceedings. 13th IEEE International Conference on, 2005, pp. 85-94.
55	K. Mohan and B. Ramesh, "Ontology-based support for variability management in product and families," in System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on, 2003, p. 9 pp.
56	S. Buhne, K. Lauenroth, and K. Pohl, "Modelling requirements variability across product lines," Los Alamitos, CA, USA, 2005, pp. 41-50.
57	T. J. Redling, "A methodology for developing new product line requirements through gap analysis," Piscataway, NJ, USA, 2003, pp. 10-1.
58	I. Hammouda, J. Hautamaki, M. Pussinen, and K. Koskimies, "Managing variability using heterogeneous feature variation patterns," Berlin, Germany, 2005, pp. 145-59.
59	M. Kim, S. Park, V. Sugumaran, and H. Yang, "Managing requirements conflicts in software product lines: A goal and scenario based approach," Data and Knowledge Engineering, vol. 61, pp. 417-432, 2007.
60	M. O. Reiser and M. Weber, "Managing Highly Complex Product Families with Multi-Level Feature Trees," in Requirements Engineering, 14th IEEE International Conference, 2006, pp. 149-158.
61	R. Rabiser and D. Dhungana, "Integrated support for product configuration and requirements engineering in product derivation," Piscataway, NJ, USA, 2007, pp. 193-200.
62	J. Savolainen, M. Kauppinen, and T. Mannisto, "Identifying key requirements for a new product line," Los Alamitos, CA 90720-1314, United States, 2007, pp. 478-485.
63	P. C. Clements, L. G. Jones, J. D. McGregor, and L. M. Northrop, "Getting there from here: A Roadmap for software product line adoption," Communications of the ACM, vol. 49, p. 1183261, 2006.
64	J. Lee and D. Muthig, "Feature-oriented variability management in product line engineering," Communications of the ACM, vol. 49, p. 1183266, 2006.
65	H. Mei, W. Zhang, and F. Gu, "A Feature Oriented Approach to Modeling and Reusing Requirements of Software Product Lines," 2003, pp. 250-256.
66	G. Bockle, "Model-based requirements engineering for product lines," in Proceedings of the first

	conference on Software product lines : experience and research directions: experience and research directions Denver, Colorado, United States: Kluwer Academic Publishers, 2000.
67	A. Hein, M. Schlick, and R. Vinga-Martins, "Applying feature models in industrial settings," in Proceedings of the first conference on Software product lines : experience and research directions: experience and research directions Denver, Colorado, United States: Kluwer Academic Publishers, 2000.
68	T. Kishi and N. Noda, "Aspect-oriented analysis for product line architecture," in Proceedings of the first conference on Software product lines : experience and research directions: experience and research directions Denver, Colorado, United States: Kluwer Academic Publishers, 2000.
69	D. Fey, R. Fajta, and A. Boros, "Feature Modeling: A Meta-Model to Enhance Usability and Usefulness," in Proceedings of the Second International Conference on Software Product Lines: Springer-Verlag, 2002.
70	S. Ferber, J. Haag, and J. Savolainen, "Feature Interaction and Dependencies: Modeling Features for Reengineering a Legacy Product Line," in Proceedings of the Second International Conference on Software Product Lines: Springer-Verlag, 2002.
71	M. Mannion, "Using First-Order Logic for Product Line Model Validation," in Proceedings of the Second International Conference on Software Product Lines: Springer-Verlag, 2002.
72	T. Kishi, N. Noda, and T. Katayama, "A Method for Product Line Scoping Based on a Decision-Making Framework," in Proceedings of the Second International Conference on Software Product Lines: Springer-Verlag, 2002.
73	D. Simon and T. Eisenbarth, "Evolutionary Introduction of Software Product Lines," in Proceedings of the Second International Conference on Software Product Lines: Springer-Verlag, 2002.
74	F. Claudia and H. Ralf, "Product line potential analysis," in Third International Conference of Software Product Lines Boston, MA, USA, 2004.
75	M. Eriksson, J. Börstler, and K. Borg "The PLUSS Approach - Domain Modeling with Features, Use Cases and Use Case Realizations," in Ninth International Conference on Software Product Lines Rennes, France, 2005.
76	T. Asikainen, T. Mannisto, and T. Soininen, "A Unified Conceptual Foundation for Feature Modelling," in Proceedings of the 10th International on Software Product Line Conference: IEEE Computer Society, 2006.
77	I. John, J. Knodel, T. Lehner, and D. Muthig, "A Practical Guide to Product Line Scoping," in Proceedings of the 10th International on Software Product Line Conference: IEEE Computer Society, 2006.
78	G. J. Chastek, P. Donohoe, and J. D. McGregor, "A Production System for Software Product Lines," in Proceedings of the 11th International Software Product Line Conference: IEEE Computer Society, 2007.
79	K. Lee, K. C. Kang, E. Koh, W. Chae, B. Kim, and B. W. Choi, "Domain-oriented engineering of elevator control software: a product line practice," in Proceedings of the first conference on Software product lines : experience and research directions: experience and research directions Denver, Colorado, United States: Kluwer Academic Publishers, 2000.
80	M. Ardis, P. Dudak, L. Dor, W.-j. Leu, L. Nakatani, B. Olsen, and P. Pontrelli, "Domain engineered configuration control," in Proceedings of the first conference on Software product lines : experience and research directions: experience and research directions Denver, Colorado, United States: Kluwer Academic Publishers, 2000.
81	S. Thiel and F. Peruzzi, "Starting a product line approach for an envisioned market: research and experience in an industrial environment," in Proceedings of the first conference on Software product lines : experience and research directions: experience and research directions Denver, Colorado, United States: Kluwer Academic Publishers, 2000.
82	J. C. Dager, "Cummins's experience in developing a software product line architecture for real-time embedded diesel engine controls," in Proceedings of the first conference on Software product lines : experience and research directions: experience and research directions Denver, Colorado, United States: Kluwer Academic Publishers, 2000.
83	M. Steger, C. Tischer, B. Boss, A. Müller, O. Pertler, W. Stolz, and S. Ferber, "Introducing PLA at Bosch Gasoline Systems: Experiences and Practices," in Software Product Lines, 2004, pp. 34-50.
84	M. Kircher, C. Schwanninger, and I. Groher, "Transitioning to a Software Product Family Approach - Challenges and Best Practices," in Proceedings of the 10th International on Software Product Line Conference: IEEE Computer Society, 2006.
85	P. Jensen, "Experiences with Product Line Development of Multi-Discipline Analysis Software at Overwatch Textron Systems," in Proceedings of the 11th International Software Product Line Conference:

	IEEE Computer Society, 2007.
86	Y. Ishida, "Software Product Lines Approach in Enterprise System Development," in Proceedings of the 11th International Software Product Line Conference: IEEE Computer Society, 2007.
87	D. Sellier, G. Benguria, and G. Urchegui, "Introducing Software Product Line Engineering for Metal Processing Lines in a Small to Medium Enterprise," in Proceedings of the 11th International Software Product Line Conference: IEEE Computer Society, 2007.
88	H. P. Jepsen, J. G. Dall, and D. Beuche, "Minimally Invasive Migration to Software Product Lines," in Proceedings of the 11th International Software Product Line Conference: IEEE Computer Society, 2007.
89	C. Tischer, A. Muller, M. Ketterer, and L. A. Geyer, "Why does it take that long? Establishing Product Lines in the Automotive Domain," in Software Product Line Conference, 2007. SPLC 2007. 11th International, A. Muller, Ed., 2007, pp. 269-274.

---

[1] 9, 11, 16, 33, 38, 40, 45, 46, 48, 50, 52, 56, 60, 61, 62, 63, 64, 66, 72, 84, 2, 3, 13, 14, 15, 18, 24, 28, 31, 49, 53, 67, 68, 70, 74, 77, 79, 80, 81, 82, 85, 86, 88, 89, 78, 69, 83, 87

[2] 1, 3, 4-14, 16-22, 25, 26, 28-46, 49-55, 57-59, 61-64, 73, 75-79, 84, 86, 87

[3] 2, 15, 23, 24, 27, 47, 48, 56, 60, 65-72, 74, 80-83, 85, 88, 89

[4] 1, 2, 8, 11, 12, 13, 31, 34, 40, 41, 43, 46, 49, 51, 53, 56, 59, 69, 70, 72, 74, 76, 77, 78, 80, 84, 85, 86, 88, 3, 24, 30, 37, 52, 87, 75

[5] 1, 2, 4-10, 13, 16-20, 22, 23, 25-29, 31-36, 39-43, 45-51, 53, 55-64, 66, 67, 69-74, 76-78, 80-86, 89, 15, 75, 3, 11, 12, 14, 21, 24, 30, 44, 88, 37, 38, 52, 54, 79, 87

[6] 9, 11, 16, 33, 38, 40, 45, 46, 48, 50, 52, 56, 60, 61, 62, 63, 64, 66, 72, 84

[7] 2, 3, 13, 14, 15, 18, 24, 28, 31, 49, 53, 67, 68, 70, 74, 77, 79, 80, 81, 82, 85, 86, 88, 89

[8] 78

[9] 69, 83, 87

[10] 3, 12, 20, 34, 43, 44, 58, 83, 90, 91, 97

[11] 3, 4, 5, 11, 14, 15, 16, 19, 24, 26, 30, 34, 37, 43, 46, 47, 48, 49, 51, 52, 53, 54, 55, 58, 65, 72, 75, 76, 77

[12] 18, 22, 33, 40, 50, 63, 64, 66, 67, 70, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89

[13] 1, 2, 6, 7, 8, 9, 10, 12, 13, 17, 20, 21, 25, 27, 28, 29, 31, 32, 35, 36, 39, 41, 44, 45, 56, 57, 59, 60, 61, 62, 71, 73

[14] 42, 68, 69, 74, 78

[15] 23, 38

[16] 1, 4, 9, 10, 15, 16, 17, 18, 19, 20, 22, 23, 28, 29, 30, 31, 32, 34, 35, 36, 38, 39, 43, 46, 47, 49, 50, 54, 57, 58, 60, 61, 62, 67, 68, 70, 74, 76, 78, 79, 80, 83, 85, 87, 89

[17] 2, 3, 5, 6, 7, 8, 11, 12, 13, 14, 21, 24, 25, 26, 27, 37, 41, 44, 45, 51, 52, 56, 59, 71, 72, 75, 81, 82, 86, 88

[18] 3, 24, 32, 49, 81, 88

[19] 38, 52, 54, 59, 67, 79, 87

[20] 11, 15, 75

[21] 1, 2, 8, 11, 12, 13, 31, 34, 40, 41, 43, 46, 49, 51, 53, 56, 59, 69, 70, 72, 74, 76, 77, 78, 80, 84, 85, 86, 88

[22] 3, 24, 30

[23] 37, 52, 87

[24] 75

[25] 1, 2, 4-10, 13, 16-20, 22, 23, 25-29, 31-36, 39-43, 45-51, 53, 55-64, 66, 67, 69-74, 76-78, 80-86, 89

[26] 15, 75