Research Article

Using Software Product Line Application in Enterprise Resources Planning Systems Systimatic Literature Review

Sabah Al.Busaidi[1] and Naoufel Kraiem[2*]

Abstract

The use of Enterprise Resource Planning (ERP) systems has dramatically increase in many companies in recent years, and research related to the implementation of ERP has increased in the last decade. On one hand, getting the advantages of ERP systems largely depends on the level of matching of ERP functionalities with the enterprise requirements. On the other hand, Product Line Engineering (PLE) is a method to manage both reuse and variability in a pre-defined way and thus brings software development to a more advanced stage. At the same time, Software Product Lines (SPLs) have emerged as a trend in software engineering. SPLs are set of software-intensive systems sharing a common, managed set of features that satisfy specific needs of a particular market or mission. Building SPLs for ERP systems will have side effects on the implementation process of ERP and will raise the flexibility of both configuration and customization issues of ERP implementation.

Objective: The main goal of this study is to provide the different ways of identifying and analyzing the techniques presented in the literature to improve ERP implementation problems using the methods, tools and techniques provided by SPLs.

Method: In order to achieve that objective, we reviewed the relevant literatures and analyzed existing studies.

Results: This literature review analyzes ten pieces of research in both ERP and SPLs concepts. It shows that the product line aspect can be used to solve configuration and customization issues of ERP.

Keywords

ERP; Software product lines; ERP configuration; ERP customization; Variability

Introduction

Reuse is employed more and more as a methodological and technological solution for developing complex software systems. A major challenge to the research community and industries is to find methods, process, appropriate language and tools for better reuse and mass configuration of software systems [1]. Software Product Line Engineering (SPLE) is a new method of software reuse. Software Product Lines (SPLs) are considered to be a successful approach to reuse in several domains like cars, printers and phones especially in

software development [2]. SPLE has been recognized as a paradigm which provides strategic software reuse [3] and it allows enterprises to reduce development costs and time to market and increase product quality [4]. It moves enterprises from developing a single software system to developing a set of software systems which share a similar and managed set of features. The use of SPL is mainly based on the reuse and mass configuration. It has been defined as a set of correlated systems that address a market segment and that are created from a common set of core assets [5].

Different Studies and research use varying definitions and implementation of ERP systems. According to Chofreh et al. [6] the ERP system combines all information and processes of an enterprise into a single system that focuses how people and organizations access, store, gather, summarize, interpret, and use information for different purposes inside the organization [6]. From the business point of view, ERP implementation has some issues related to configuration and customization, which we aim to solve by using the concept of SPLs. There are different approaches and frameworks in the literature targeting both ERP and SPL as a solution perspective. Research on ERP systems is increasing in order to adopt new trends of technologies and to change business process. Implementing an ERP system successfully is important for the future competitive strategy of a company. In 2016, Nasr and Gheith defined an ERP implementation as the process of transforming a standard ERP product into an operational system inside an organization [7]. The two basic concepts of ERP which require more consideration are configuration and customization [8]. Configuration is a normal step in any software process that does not require any change in the source code. Customization means the process of adding new features or modifying the existing features of an ERP product which require change in the source code, and deeply understanding of existing program's functionality. For instance: modifications of user interfaces, output and messages and even program codes [9]. Mazo and Assar, stated that the difference between ERP systems and PLE concur on two concepts: variability management and the capability to be configured, customized and adapted to possibly various environments [10]. The systematic handling and the reuse of artifacts of variability provide significant concepts for various industrial settings. Variability models are a key concept in software product line engineering and also ERP systems [11].

Variability in ERP systems is implemented by representing organizational data in operational tables and configuration parameters in strategic tables describing varying operational information [10]. SPL is usually represented by means of a Product Line Model (PLM). A PLM represents, in an intensive way, the collection of products that belong to the product line [12]. The goal of this paper is to combine the concepts of ERP and SPL available knowledge. As the level of importance of these topics increases, there is not enough research to cover them. Thus, the available literature is scarce: only ten papers fulfilled the study and inclusion criteria, and amount of these contributions are restricted to a limited number of methods, models and techniques.

This paper is structured as follows. Section 2 briefly presents the background, which discusses the two main concepts needed to understand the rest of the paper. Section 3 shows an overview of the

research method that was employed to perform the literature review. Section 4 highlights and discusses the main findings derived from the analysis of the results and points out the recommendations and Section 5 explains these results. Finally, we conclude the paper and highlight possible future work in Section 6.

Background

Product Line Engineering (PLE) is an emerging model that allows the development products by reuse of artifacts from a product line. A Product Line (PL) is a set of system sharing a similar, controlled set of features that satisfy the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a predefined way [12]. Software Product Line Engineering (SPLE) often distinguishes two main processes. The first is a domain engineering process which deals with developing and maintaining reusable core or domain assets that typically are reusable components of software. In this process, products are built from a set of artifacts that have been specifically designed as a reusable core asset base. These core assets comprise the software architecture, its documentation, specifications, tools and software components [10]. The second process is application engineering which deals with the development of software products, or applications, using these core assets for quick and efficient composition of software products tailored to the customer's requirements [13]. The transition between the two domains is made through a configuration that adapts a domain model to define an application product [14]. SPL is usually shown by using a Product Line Model (PLM) [12]. Product Line Models (PLMs) can be specified with different languages like Feature Models, Constraint-based PLMs, or Orthogonal Variability Models and Decision Models [15,16]. Therefore, PLE is an important method and is used to reduce time to market and TCO cost as well as it reducing the configuration time of new products [17]. According to Voelter's research, there are some reasons for adopting SPL in enterprises like, it minimizes development time, effort, cost, and complexity by taking advantage of the commonality within a portfolio of same products [18]. Beside obvious advantages of SPL, there are also several potential drawbacks and challenges. According to Schaefer's findings [19], reducing the time of software development affects the product's quality. This is especially occurs in the development of applications with high safety and security requirements. Moreover, maintenance of product lines is more difficult and expensive compared with single-system maintenance [19]. Variability is one of the main keys of SPLs. It describes the variances that exist across the artifacts like various implementations or algorithms for different environments and/or requirements [20]. Different variability models and techniques have been proposed in [21], for instance: MAP Model, Feature Model and OVM Model. Voelter, has defined the activity that is concerned with identifying, designing, implementing, and tracing flexibility in SPLs as variability management [18]. Furthermore, Triki [12] has stated that variability is the ability of a system or artifact to be configured, customized, extended, or changed for use in a specific context. The term customization refers to the process of adapting a software product to a range of different users by changing features in the software in order to satisfy their needs. Capturing customization opportunities is known as variability points which means an important activity that allows developers to set the effective ways in which software artifacts can be reused [5]. According to Heidenreich [22], there are several steps in order to model variability in the solution space for example: create a reference model for the family of products in the software product line. This reference model must incorporate certain variability mechanisms for supporting the variants specified in the feature model. In recent years, ERP systems have received much attention. From the ERP perspective, Lotfy (2015), defined ERP systems as architectural models which identify the technology, processes, and the people as the core components of the ERP environment [23]. Enterprise resource planning (ERP) system implementation projects are very complex, costly and have a high failure risk. There are two strategic approaches of ERP system implementation. The first approach is the TEIs go for a generic version of ERP system. In this approach the institute has to make the changes in the processes to fit the functionality of the ERP system. This takes advantage of future upgrades in the system. The second approach allows institutes to customize the ERP system as per the processes of the institute. ERP implementation focuses on two main challenges: configuration and customization [24]. It is important to understand the configuration and customization aspects of ERP system in order to manage the systems efficiently. According to the study conducted by Misfer in 2015, the customization process requires source code modification, while configuration is applied through a predefined parameter setting to modify application functions within pre-defined scope [25]. ERP configuration is about balancing the way the customer wants the system to work, for example; customer requirements, with the way it was designed to work. ERP systems typically build many changeable parameters that modify system operation [10]. The configuration process is considered as one of the key successes of any SaaS ERP, while customization of ERP is one of the main issues that organizations have been complaining about due to its cost and complexity [25]. Several different studies have been conducted to identify the benefits gained by successful ERP implementation in enterprises such as Chen and Wang's study in 2016, which argues that the potential benefits of ERP implementation can be classified into tangible and intangible benefits. Despite these significant benefits, there are some major problems with ERP systems [26]. Furthermore, in 2001, Toni and Klara summarized some drawbacks in the ERP system that increase the cost of implementing software packages, the cost of software incremental hardware, training, and implementation support is also high [27]. Even so, the complexity of ERP systems is the most important problem if the systems are not used in an efficient and predictable way. For instance, an ERP system implementation can takes one month to five years [23].

Research Method

This section discusses the methodology followed in collecting and analyzing the research and journals used in this report. A main element in the systematic literature reviews is the clear definition of a review protocol in the planning phase that guides its execution. It tries to minimize researchers' bias and helps in structuring the retrieved results. The protocol describes: the research questions for the literature review, the search strategy; the search strings and terms used for searching; the selection and quality assessment criteria that are general restrictions, the inclusion and exclusion criteria for selecting a relevant subset of the publications found, and the data extraction process. It is somewhat difficult to narrow the report on ERP and SPL to specific orders; the relevant material is spread out across numerous studies. The work reported in this paper was achieved by using Kitchenham's et all methodological strategies. Figure 1 is an overview of the main steps of the research process [10]. It provides a discussion of the development of the review protocol used in this study. The review protocol identifies the methods used for finding primary studies [1].

Three main stages can be followed in order to perform a systematic review: planning the review, conducting the review and reporting the results as shown in Figure 2 [10].

The first phase (Planning Review) includes identifying the basic needs for a review and developing a review protocol.

The second phase (Conducting Review) involves identifying and selecting main studies, evaluating the quality of the primary studies, and extracting and producing the information reported in the primary studies [1].

The third phase (Reporting Review) comprises synthesizing the data removed during the review execution and summarizing the results of the involved primary studies. The results and analysis of this review is reported in Section 4. The extracted information are presented according to the review questions and is supported by descriptive analysis and graphical representation [1].

Literature Review Conduct and Results

Systematic reviews are becoming a standard research method amongst software engineers. The lack of explored topics holds true in the areas of Enterprise Resource Planning (ERP) and Software Product Lines (SPLs) and justifies the need for more systematic literature reviews of software product line methods, techniques, and the approaches used to improve ERP systems. Systematic reviews are conducted to summarize existing evidence concerning a specific research question, topic area and to identify areas where research is lacking. The research questions asked in this literature review were designed to summarize what software product line methods, techniques, and approaches exist today, how we can use them in solving ERP issues and what methods or approaches are proposed in the study.

Research questions

Our review was guided by the following research questions:

RQ.1 What is the research proposed (method, approach or framework)?

RQ.2 At which stages are SPL tools and techniques used to handle configuration and customization?

RQ.3 Which variability model is used to deal with configuration and customization?

RQ.4 Who are benefits the most from the presented work? (Supplier, Partner Company, end user,)?

RQ.5 Which support tool is developed to automate software product line application in the system?
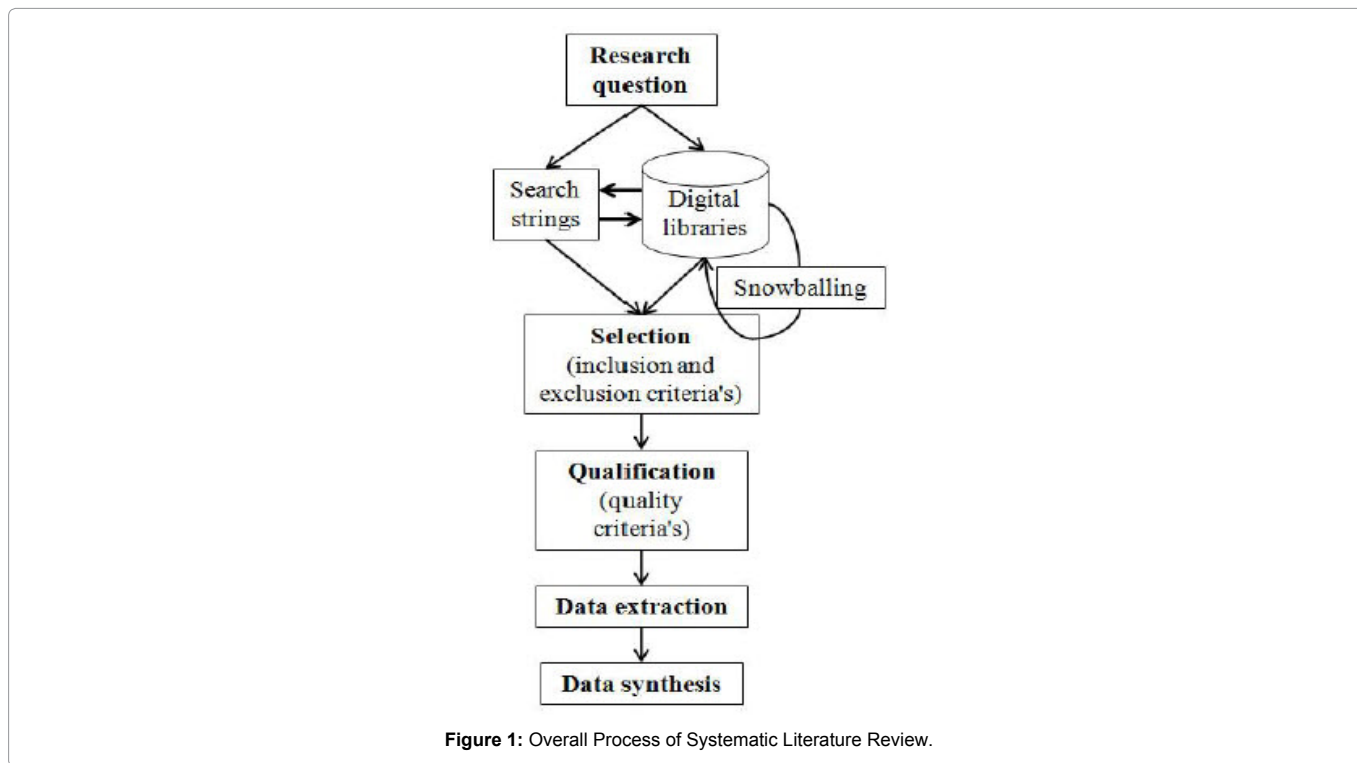

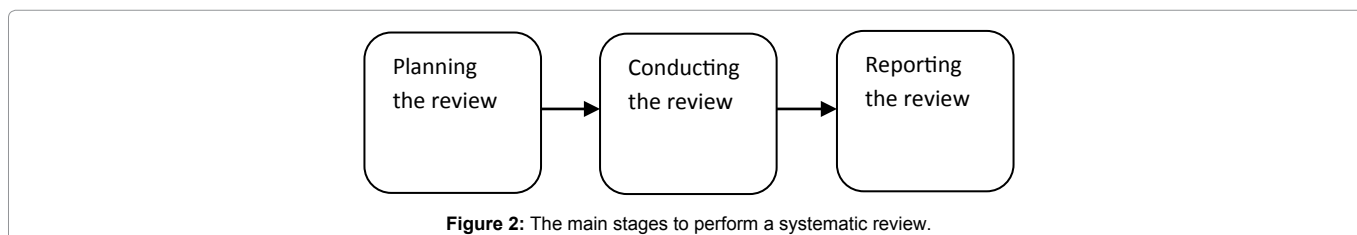Figure 1: Overall Process of Systematic Literature Review.

Figure 2: The main stages to perform a systematic review.

RQ.6 How is the proposed method, approach or framework applied and validated?

Search strings and digital libraries

Specific publication databases were selected on the basis that they included research papers in computing and information technology sciences. These digital databases were chosen because they offered the most essential and highest impact journals and conference proceedings and covered the fields of enterprise resource planning and software product line engineering [1]. We referred in the search string to the title and the abstract of the paper and we defined the following search strategy: the sources were selected based on an analysis of product line and ERP domain literature. The presented review includes studies published from 2006 to 2016 considering issues related to ERP implementation that are configuration and customization and software product line techniques. We used Google Scholar, Scopus and the digital search feature of IEEE. Retrieving studies and papers from the above databases requires a specific combination of keywords. These keywords establish the identity of papers that suggest the integration of Enterprise Resource Planning and Software Product Line.

We have used the following search terms used to identify possibly relevant publications in the computing and information technology sciences literature, "Software product line" + "ERP" or "product line engineering" + "enterprise resource planning", or "Enterprise Resource Planning" + "Software Product Line Engineering" or "Software Product line models" or "SPL variability", "ERP", "ERP implementation", "ERP configuration", "ERP customization".

Selection and qualification

As the keywords used to find researches are indicated above, the following section lists the exclusion criteria which are defined in the selection stage to decide the papers appropriate for the study:

1. **Publications published between 2006 and 2016 are kept**: Only interested in current and relevant publications conducted in recent years. The year 2006 is mentioned up to the year of 2016

2. **The paper should be already published in a trusted conference, journal, report or workshop:** Tutorials and electronic books are excluded.

3. **Only publications written in English were kept**: language barriers restrict this review to consider papers written in English only, for example, papers written in Dutch or in Spanish were excluded.

4. **Papers where the full-text is available:** If the full-text is available for review it is included, otherwise there is no information to review and extract.

At the end of these stages, only 10 studies were left and they are presented in Appendix 1.

According to our research, the earliest experience was in 2007 by Software Product Lines Approach in Enterprise System Development [28]. There are no articles published between 2000 and 2006 selected in this review.

Data extraction

We read each of the 10 papers' abstracts very closely to extract the required data. We used a predefined form which consists of a number of attributes to extract and store the data. These attributes are required in order to answer the main review questions [29]. In addition to obtaining the information required to answer the research questions and quality assessment criteria, the following standard information was also extracted from each primary study: Title of the Paper, Sources (Database and Journal), Date Published, Paper URL, Document Object Identifier (DOI) and Authors. The main goal of collecting the above information is to provide analysis of the meta-data of the research itself. This measurement will provide insight into the growth and interest in software product line research. However, this review has limited its work to reporting the findings associated with answering the research questions [1].

Table 1 summarizes the different methods, approaches or frameworks that exploit SPL techniques in ERP system, configuration and customization processes. These methods, approaches or frameworks are distinct in the aspects which are covered, loosely structured and not systematically anticipated and their descriptions vary considerably. Each author has a different vision of the ERP and SPLs and the authors present their works accordingly.

Table 2 summarizes the implementation stages at which SPL techniques are applied whither in configuration or customization or both. It is clearly shown in the table that configuration is the most implementation stage used by SPL techniques. Martinez's study [30] mentions, that configuration and customization are considered as being the same.

The variability models and modeling tools which have been used in each method, approach or framework are presented in Table 3. There is no clearly mentioned tool in most of the studies except the one proposed by Wolfinger which mentions DOPLER as a tool that is used for variability modeling in the ERP engineering context [31].

Table 1: Data Extraction for RQ1: Proposed Method, Approach or Framework.

Paper	Proposed Method Approach or Framework
P1	• Describes what application centric architecture is and how this approach differs from others with lessons learned for the last five years
P2	• Discusses the integration of a plug-in platform for enterprise software with an existing product line engineering tool suite to support system adaptation
P3	• Proposes a Variant Description Model that comprises all variants resolved and based on the variability defined in the feature model
P4	• Presents Mapping between the feature model and the family model, which contains ERP configuration options and documentation.
P5	• Deducts a Variant Result Model which means the concrete product configuration
P6	• An investigation of PLE role in identifying and codifying tacit business knowledge in two industrial case studies in the domain of ERP system
P7	• A study of a particular software product line which used ERP as experimentation
P8	• Discusses ERP domain constraints and provides conceptual solutions on how to adapt and extend SPL techniques for this particular context. • Aims to support product configuration in software ecosystems based on several variability models with different semantics that have been created using different notations.
P9	• Introduces practical experience from the application of PLAs in four enterprises of the ERP system domain • Uses decision-flow forms as a variability resolution process which involved of a set of interrelated decisions for a suitable ERP configuration
P10	• Introduces SPLs requirements elicitation approach for cloud ERP systems.

Table 4 shows the actors who are concerned in the existing variability model. End-users are the most implicated actors in the configuration and customization processes.

Table 5 describes the tools suggested to support the proposed method, approach or framework and the variability modeling approach. Each study presents different tools and techniques in order to support variability modeling. A study published by Wolfinger [31], proposes complete configuration tool suites to support variability modeling and to prepare and direct product derivation and customization.

Table 6 presents the case studies in which the proposed method, approach or framework is validated and evaluated with the final results of the published work.

Discussion

A systematic literature review was conducted to identify the different ways of applying SPLs engineering to ERP systems. Many studies have mentioned various definitions and implementation process of ERP and SPLs aspects. The review highlights several of the approaches, methods, techniques and tools needed for combining SPLs concepts with ERP systems. A considerable amount of research addresses SPLs and ERP concepts, but far fewer publications cover them both. From the business point of view, ERP implementation has some issues related to configuration and customization. As most research focused on the aspects of ERP system, there are omissions in how to solve these issues. According to the literature, the product line approach was used to either configure or customize ERP systems, or both, by using several methods with different approaches. For example Hamza and Aly in [32], Nöbauer in [11], Nasir in [33], and Galindo in [16], have taken the advantages of software product line just in ERP systems configuration. Likewise Wolfinger et al [31], Leitner [34], Ouali [35], and Ali [7], are concerned with configuration and customization of ERP systems. On the other hand, Martinez and Alonso [30] claim that there is no differentiation between ERP configuration and customization (RQ2). For (RQ3), there is no clearly mentioned tool in most of the studies except for [31], which mentions DOPLER as a tool that is used for variability modeling in an ERP engineering context. Moreover, for (RQ4), the End-users are the most implicated actors in the configuration and customization processes [28,30,31,32,35]. Furthermore, each study presents different tools and techniques in order to support variability modeling. Wolfinger's [31], study proposes a complete configuration tool suite to support variability modeling, that is a plug-in feature model to customize the ERP system and support system adaptation at runtime, while Leitner and Kreiner [34], represent business process features and configuration features in a two-layer FODA model in order to manage different ERP configuration variants.

Conclusion

The goal of this paper was to identify the different ways to apply the software product line application to ERP systems. To recognize this, a systematic literature review was conducted by following a search strategy and applying selection criteria and a qualification process. In the data extraction process, we found that this approach has been applied for the main ERP implementation processes, configuration and customization. The literature, however, shows the importance of product line engineering methods, techniques and tools, but there is still a lack of interest in addressing ERP engineering issues with the product line strategy. Our iterative method resulted in a final set of six different studies from which we extracted relevant

elements for six research questions: (1) The ERP implementation issues, (2) the proposed method, (3) the variability model used and artifacts presented in the model, (4) the actors who benefit from the method (5) the tools, and (6) the case study used. Based on selected papers selected from the literature we noticed that product line engineering could reduce the complexity of the ERP configuration and customization issues. This item was analyzed and discussed and two scenarios have been presented to demonstrate its feasibility in accordance with our SLR results.

Table 2: Data Extraction for RQ2: Implementation Stages.

Paper	Implementation Stages (Configuration & Customization)
P1	• ERP configuration and customization
P2	• ERP configuration and customization
P3	• ERP configuration = ERP customization
P4	• ERP configuration = ERP customization
P5	• ERP configuration
P6	• ERP configuration and customization
P7	• ERP configuration
P8	• ERP configuration
P9	• ERP configuration
P10	• ERP configuration and customization

Table 3: Data Extraction for RQ3: Variability Model.

Paper	Variability Models And Modeling Tools
P1	• Three variability management • Variability Management of Platform • Variability Management of Non-Functional Requirements • Variability Management of Functional Requirements
P2	• Feature model • DOPLER tool for • variability modeling • high level decision making
P3	• Variability modeling tool not clearly mentioned • Variations points are ERP functionalities according to user's requirements
P4	• Two-layer feature model 'FODA' • First layer: business processes features • Second layer: configurations features for specific customers
P5	• Feature model • Variability modeling tool not clearly mentioned
P6	• Variability modeling tool not clearly mentioned • Feature model
P7	• Variability modeling tool not mentioned
P8	• Feature model a centralized feature repository
P9	• Feature model • OVM-style model • Decision-oriented model
P10	• Feature model

Table 4: Data Extraction for RQ4: Actors Involved.

Paper	Actors Involved
P1	• End users (clients)
P2	• End users (stakeholders)
P3	• End user and Partner company
P4	• Partner company
P5	• End users and Partner company
P6	• End users and Partner company
P7	• Partner company
P8	• Partner company
P9	• Stakeholders, Vendors and suppliers
P10	• Stakeholders, requirements elicitation system, requirements engineers, development engineers, and test and integration engineers.

Table 5: Data Extraction for RQ5: Variability Tool Support.

Paper	Variability Tool Support
P1	• Variability tool support not clearly mentioned • relational database (RDBMS) software as middleware open source software (OSS) • Inversion of Control (IoC) container as basic mechanism of pluggable architecture. • Use of a SPL platform as middle layer to manage variability
P2	• NET-based plug-in platform for dynamic loading, unloading and composition of components • Decision-king employed to define the variability model • Project-King that allows defining scenario specific configurations of PL variability model • Configuration-Wizard which processes the scenario specific variability model and presents decisions to the user in a wizard-like interface
P3	• Product Line Unified Modeler (PLUM) tool suite used to model a specified scope in each ERP product, implementation and management of Software Product Lines (SPL
P4	• PURE: variants, the tool used, shipped as Eclipse plugins. • Three plugins: • A model validation plugin that enforces a domain specific internal structure of the feature model • An import plugin that extracts customization menu items in order to build up the Family Model in an automated way • A transformation plugin to set the ERP customizing parameters according to the Variant Result Model
P5	• Product Line Unified Modeler (PLUM) to improve the capability of the companies to identify and encode tacit business knowledge into reliable, easy to use and to use share knowledge architecture.
P6	• Not clearly mentioned
P7	• PL4X ERP configurator • Microsoft dynamic AX • ALM for the establishment of traceability between the problem and solution spaces
P8	• Artificial Intelligence techniques • Decision tree • Rapid Miner tool which allows user to experiment with a host of decision tree algorithm
P9	• Application programming interface (the Invar API) to support different legislation strategies for variability models • FaMa or FAMILIAR tools for variability (feature) modeling
P10	• Code generator tool which is a development tool that takes as an input the features configurations and builds the reusable assets based on these configurations

Table 6: Data Extraction for RQ6: Case Study and Results.

Paper	Case Study and Results
P1	• Defined the boundary between application and platform based on the application development perspective at first, they built up the platform by integrating dozens of OSS products to boost our SPL projects. • The in-Motion pattern works well in this case, which allows the gradual evolvement of the architecture and the designs through validation in actual projects.
P2	• Conducts a case study in collaboration with the industrial partner BMD Systemhouse GmbH • Develops 5 different and advanced usage scenarios • Shows the feasibility and usefulness of the approach by means of these usage scenarios where the variability of the ERP system is represented by means of variability models. • The different elements of these variability models represent modules of the ERP system and the relationships among these modules
P3	• The experience was done in the Reuse-Cluster Approach Project with four ERP major companies in Egypt
P4	• Applies the method in three European divisions of a metal forming company. Each company uses an SAP ERP system. o Describes 3 scenarios • Isolated ERP solution development • "Template"-based solutions • PL solutions o Presents quantitative analysis and Lessons learned to prove the feasibility of applying SPLE concept to ERP system
P5	• Conducts two industrial case studies within a Cluster Approach Project implemented in four major companies in Egypt in the domain of ERP system.
P6	• Evaluates both Copiere and Openbravo.
P7	• Provides Concrete examples from Microsoft dynamic AX platform • Support for sales consultants through sales scenario • Support for customer application configuration through Analysis and Design Scenario • Implement prototype tool called PL4X ERP configurator.
P8	• In order to measure the prediction accuracy of the tree on product configurations currently in progress, they have used Rapid-Miner to test the tree with five such configurations. When they compared the results, they find that a correct prediction of a faulty configuration was made in 2 out of 3 cases
P9	• The feasibility of the approach and its implementation are shown by using them with the most three common types of variability modeling approaches in the PL community. • The example presented in the paper is derived from industrial experience in ERP, to validate the feasibility and flexibility of the approach. They further applied the approach to support the configuration of privacy settings in the Android ecosystem based on numerous variability models. • The performance of different model enactment strategies used in the approach is evaluated.
P10	• This ERP system has been implemented in the National Research Center (NRC) (further details unavailable).

Reference

1. Ahnassay A, Bagheri E, Gasevic D (2011) Empirical Evaluation in Software Product Line Engineering 1-52.

2. Ouali S, N Kraïem, Z Al-khanjari, Y Baghdadi (2013) A Model Driven Software Product Line Process A Meta-model for SPL. Journal of Software 10: 447-454.

3. M Acher, P Collet, P Lahire, S Moisan, JP Rigault (2011) Modeling variability from requirements to runtime. 16th IEEE Int Conf Eng Complex Comput Syst.

4. S Kato, N Yamaguchi (2011) Variation Management for Software Product Lines with Cumulative Coverage of Feature Interactions. 15th International Software Product Line Conference Munich, Germany.

5. A Khtira, A Benlarabi, B El Asri (2015) A Tool Support for Automatic Detection of Duplicate Features during Software Product Lines Evolution. IJCSI 12: 1-11.

6. AG Chofreh, FA Goni, AM Shaharoun, S Ismail, JJ Klemeš (2014) Sustainable enterprise resource planning: Imperatives and research directions. J Clean Prod 71: 139-147.

7. M Ali, ES Nasr, MH Gheith (2016) A Requirements Elicitation Approach for Cloud Based Software Product Line ERPs. 2nd Africa Middle East Conf Softw Eng Cairo, Egypt.

8. Jelliti M, Sibilla M, Jamoussi Y, Ben Ghezala H (2010) A Model Based Framework Supporting ITIL Service IT Management. EMMSAD'2010, Hammamet, Tunisia 50: 208-219.

9. A Amid, M Moalagh, A Zare Ravasan (2012) Identification and classification of ERP critical failure factors in Iranian industries. Inf Syst 37: 227-237.

10. R Mazo, S Assar, C Salinesi, N Ben Hassen (2014) Using Software Product Line to improve ERP Engineering : Literature Review and Analysis.

11. M Nöbauer, N Seyff, R Stoiber (2012) Managing Variability of ERP Ecosystems Research Issues and Solution Ideas from Microsoft Dynamics AX. 6th International Workshop on Variability Modeling of Software-Intensive Systems, Leipzig, Germany.

12. R Triki, C Salinesi, R Triki, C Salinesi, R Mazo (2015) Three strategies to Specify Multi-Instantiation in Product Lines. 9th International Conference on Research Challenges in Information Science.

13. G K Hanssen (2010) Opening Up Software Product Line Engineering. ICSE Workshop on Product Line Approaches in Software Engineering, Cape Town, South Africa.

14. Uzma Afzal, Tariq Mahmood, Zubair A. Shaikh (2016) Intelligent software product line configurations A literature review. Computer Standards & Interfaces 48: 30-48.

15. L Jaramillo, C Salinesi, L Jaramillo, C Salinesi, C Dumitrescu (2013) Material Needs Forecast for Product Lines a Bayesian-based Analysis Approach. 25th International Conference on Software and Systems Engineering and their Applications, France.

16. A Galindo, D Dhungana, R Rabiser, D Benavides, A. Galindo, et al. (2015) Supporting Distributed Product Configuration by Integrating Heterogeneous Variability Modeling Approaches. Information and Software Technology 62: 78-100.

17. R Mazo, C Salinesi (2008) Methods Techniques and Tools for Product Line Model Verification.

18. M Voelter, I Groher (2007) Product line implementation using aspect-oriented and model-driven software development. 11th Int Softw Prod Line Conf, Kyoto, Japan.

19. I Schaefer, R Hähnle (2011) Formal Methods in Software Product Line Engineering. Software Technologies.

20. D Benavides, D Schmidt (2013) Evolving feature model configurations in software product lines. Journal of Systems and Software 87: 119-136.

21. Ouali S, Kraïem N, Ben Ghezala H (2015) Intentional Software Product Line using Model Driven Engineering Intentional Software Product Line using Model Driven Engineering. Int J Emerg Technol Adv Eng 5: 447-454.

22. Heidenreich F, Sánchez P, Santos J, Zschaler S, Alférez M, (2010) Relating Feature Models to other Models of a Software Product Line: A Comparative Study of Featuremapper and VML. Lec Notes Comput Sci 6210: 69-114.

23. Lotfy MA (2015) A Conceptual Model to Measure ERP User-Value. Issues Inf Syst 16: 54-63.

24. Parthasarathy S, Sharma S (2014) Determining ERP Customization Choices using Nominal Group Technique and Analytical Hierarchy Process. Comput Ind 65: 1009-1017.

25. Al-Shardan MM, Ziani D (2015) Configuration as a Service in Multi-Tenant Enterprise Resource Planning System. Lect Notes Softw Eng 3: 95-100.

26. Shen Y, Chen P, Wang C (2016) Computers in Industry a Study of Enterprise Resource Planning (ERP) System Performance Measurement using the Quantitative Balanced Scorecard Approach. Comput Ind 75: 127-139.

27. Somers T, Nelson K (2001) The Impact of Critical Success Factors Across the Stages of Enterprise Resource Planning Implementations. Proc 34th Annu Hawaii Int Conf Syst Sci 8: 1-10.

28. Ishida Y (2007) Software Product Lines Approach in Enterprise System Development. Int Softw Prod Line Conf.

29. Chen L, Babar MA, Ali N (2009) Variability Management in Software Product Lines : A Systematic Review. Int Softw Prod Line Conf 81-90.

30. Martinez J, Alonso C (2016) Introducing Product Line Architectures in the ERP Industry: Challenges and Lessons Learned. SPLC Work 263-266.

31. Wolfinger R, Reiter S, Dhungana D, Grünbacher P, Prähofer H (2008) Supporting Runtime System Adaptation through Product Line Engineering and Plug-in Techniques.

32. Hamza H, Aly G (2010) Using Product Line Architectures to Leverage Systematic Reuse of Business Knowledge: An Industrial Experience. Knowledge-Oriented Prod Line Eng.

33. Nasir SZ, Mahmood T, Shaikh MS, Shaikh ZA (2015) Fault-tolerant Context Development and Requirement Validation in ERP systems. Comput Stand Interfaces 37: 9-19.

34. Leitner A, Kreiner C (2010) Managing ERP Configuration Variants : An Experience Report.

35. Ouali S, Kraiem N, Ben Ghezala H (2011) Framework for Evolving Software Product Line. Int J Softw Eng Appl 2: 34-51.

Author Affiliation   Top

[1]Computer Science Department Sultan Qaboos University Muscat, Oman

[2]Oman RIADI- ENSI Campus of Manouba, Tunisia