Managing variability of ERP ecosystems: research issues and solution ideas from Microsoft Dynamics AX

Nöbauer, Markus ; Seyff, Norbert ; Dhungana, Deepak ; Stoiber, Reinhard

Abstract: Systematic reuse of artifacts and a clear understanding of the variability within a product family are key success concepts within diverse industrial domains. Nevertheless, there are still many open issues regarding adapting and tailoring of software product line engineering approaches to specialized domains. The nature of ERP systems would suggest the application of product line techniques, but the limitations and constraints within this domain makes this a challenging task from the viewpoint of a partner company. In this paper we discuss ERP domain constraints and provide first conceptual solutions on how to adapt and extend software product line techniques for this particular context. Furthermore, we present a first tool prototype to support sales consultants at ERP partner companies.

DOI: https://doi.org/10.1145/2110147.2110150

Managing Variability of ERP Ecosystems: Research Issues and Solution Ideas from Microsoft Dynamics AX

Markus Nöbauer
InsideAx GmbH
Lunzerstrasse 64
A-4031 Linz
markus.noebauer@insideax.at

Norbert Seyff
University of Zurich,
Requirements Engineering Research Group,
CH-8050 Zürich
seyff@ifi.uzh.ch

Deepak Dhungana
Siemens AG Austria
Corporate Technology
A-1211 Vienna
deepak.dhungana@siemens.com

Reinhard Stoiber
University of Zurich,
Requirements Engineering Research Group,
CH-8050 Zürich
stoiber@ifi.uzh.ch

ABSTRACT
Systematic reuse of artifacts and a clear understanding of the variability within a product family are key success concepts within diverse industrial domains. Nevertheless, there are still many open issues regarding adapting and tailoring of software product line engineering approaches to specialized domains. The nature of ERP systems would suggest the application of product line techniques, but the limitations and constraints within this domain makes this a challenging task from the viewpoint of a partner company. In this paper we discuss ERP domain constraints and provide first conceptual solutions on how to adapt and extend software product line techniques for this particular context. Furthermore, we present a first tool prototype to support sales consultants at ERP partner companies.

Keywords
ERP Ecosystems, Product Line Engineering, Variability management, Microsoft Dynamics AX

1. INTRODUCTION
Software product line engineering has long been described, adopted and reflected upon, as a promising approach for dealing with families of similar products [1]. The reuse of artifacts and the systematic handling of variability provide important concepts for diverse industrial settings [2]. Nevertheless, there are still many open issues when it comes to adapting and tailoring product line approaches and tools to specialized domains, in order to fit domain-specific processes and technological settings. In this paper we describe issues of introducing software product line and variability concepts in the enterprise resource planning (ERP) domain, which poses special challenges related to variability management beyond organizational borders. We use concrete examples from Microsoft Dynamics AX[1].

VaMoS '12, January 25-27, 2012 Leipzig, Germany

ERP systems support companies and public organizations performing and monitoring business activities. Microsoft Dynamics AX is an integrated business management solution for medium and large enterprises, which covers multiple business areas in one product. Software distribution is shared between Microsoft and Partner companies (see Figure 1). Microsoft maintains the standard software product (i.e., the platform, which resembles a software ecosystem [3]), develops and integrates solutions to support strategically relevant business areas and releases new versions of the Dynamics AX ERP platform. Partner companies sell the ERP system to customers and customize the system to fit to customers' needs (see Figure 1). Furthermore, partner companies develop vertical solutions to address new or additional requirements within a specific business area.

The nature of ERP systems would suggest the introduction of product lines techniques, because most tailoring and customization activities recur quite frequently. However, so far, Microsoft has not considered applying a systematic product line approach. A particular partner company, on the other hand, has limited interest in building and maintaining a product line for Dynamics AX. First, partner companies have limited resources to do so and every new release of the Dynamics AX ERP platform would outdate the product line. And second, since Dynamics AX essentially resembles a software ecosystem, there exist particular limitations, resulting from the lack of code ownership and similar, cf. section 3. However, partner companies would still benefit from introducing product line approaches that allow a more rapid, higher-quality and less costly development of prototype systems to support sales and early-phase requirements activities.

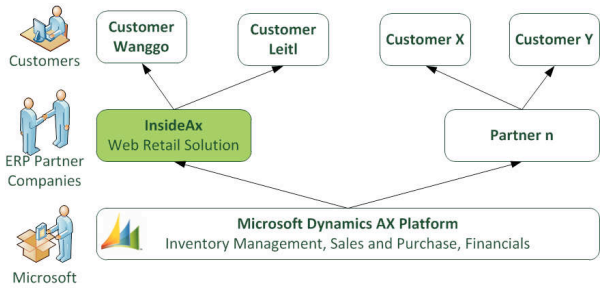

Figure 1 ***Typical organization of vendor, business partners and customers, reflecting the ecosystem nature of MS Dynamics AX***

[1]http://www.microsoft.com/en-us/dynamics/erp-ax-overview.aspx

In this paper, we present first results from a long-term industrial research project, from the perspective of a partner company in a software ecosystem. Section 2 presents the research goals and objectives. Section 3 discusses ERP domain-specific constraints that limit the introduction and adaption of software product line techniques. In Section 4 we highlight early steps in an ERP project and depict a vision of how we plan to support these steps. In addition to a conceptual solution we also present an early tool prototype in Section 5. In Section 6 we discuss related work. And in Section 7 we highlight our contributions and conclude the paper.

2. RESEARCH GOAL AND OBJECTIVES

The goal of the PL4X research project (Product Lines for AX), as described in section 4, is to make software product line techniques available and useable in the ERP domain. The results of the research project should – in general – enable partner companies to benefit from product line approaches in order to reduce time and costs in ERP projects.

We foresee that particularly early steps in ERP projects will benefit from software product lines. This will lead to time and cost savings in ERP projects, with appropriate support. We have set the following objectives:

Objective 1 *– Foster the reuse of configuration parameters:* A critical step in starting an ERP project is the development of a sales prototype, which reflects high-level customer requirements. The quality and accurateness of such a prototype are key criteria for a customer to award a contract to an ERP partner. Currently ERP partners spend a lot of effort in building sales prototypes, which might be wasted if a competitor wins the tendering procedure. Objective 1 aims at structured reuse of configuration parameters, which significantly automates and eases the generation of sales prototypes.

Objective 2 *– Foster the reuse of customer requirements:* Although every customer has specific requirements, we have experienced that there is a significant overlap between the requirements of customers from a particular domain (e.g., construction industry). Objective 2 aims at identifying and communicating potentially overlapping customer requirements, which can also be relevant for a new customer.

Objective 3 *– Foster the reuse of features:* The identification of relevant variable features based on customer requirements is a challenging process. When features are not correctly identified this might lead to multiple implementations of the same functionality for different customers. Furthermore, the manual identification of variable features is time consuming and error prone. Objective 3 aims at fostering and automating the identification of relevant features and easing their reuse.

Research in the field of software product lines has developed several approaches that would provide solutions to the described objectives, e.g. [4]. However, existing limitations in the ERP domain limit the applicability of these solutions.

3. ERP DOMAIN LIMITATIONS

As a first step we have identified key domain limitations and constraints from analyzing the Dynamic AX ecosystem (see Figure 1) and conducting interviews and discussions with consultants from ERP partner companies. The nature of the Microsoft Dynamic AX ecosystem raises several limitations for adapting and tailoring state-of-the-art product line approaches. These key limitations include:

Distributed Code Ownership: State-of-the-art software product line engineering approaches, e.g. [4], assume that the organization applying software product lines owns the code and therefore can actively control changes and domain artifacts. In the case of Microsoft Dynamics AX it is Microsoft who owns the code and partner companies have to accept changes. While from Microsoft's viewpoint this resembles a software ecosystem [3], these concepts cannot be applied by a partner company, since it does not own the ERP system code. Partner companies have hardly any influence on new releases, apart from their own vertical solutions that are built upon specific versions of the ERP system.

Documentation of variability: Variability models are a key concept in software product line engineering. In the case of Microsoft Dynamics AX no variability model (e.g. orthogonal variability model [4] or feature model [8]) is available or announced. Rather, a large and comprehensive set of configuration keys is delivered, which is very fine-grained and would need to be mapped to a more abstract variability model. For partner companies it is practically impossible to establish and maintain such a model, because of the size and complexity of the ERP systems. Furthermore, this model would be invalidated with every new release of Dynamics AX.

Variability tool support: Microsoft provides a wide range of tool support for software development, which can also be used in ERP system development. However, as the ERP system vendor does not address the documentation of more abstract variability models, the required tool support to establish and maintain an ERP software product line is missing, too. This tool infrastructure would need to handle the variability model on the one hand and the ERP system's available parameters and custom-developed vertical solutions on the other.

Interdisciplinary work: ERP projects are highly interdisciplinary; the ERP consultant often has limited technical knowledge, but is a specialist for a particular business domain or process. However, we also consider ERP consultants to be potential users of envisioned ERP product line tools. This would mean that different views have to be possible on the variability model. ERP consultants, for example, may require tools that are able to generate quite abstract views on the variability model that is being tailored. And developers, for example, may require much more fine-grained views on the variability model and configuration parameters. Dynamic view generation from an integrated model or disciplined traceability will be required to ensure consistent views on these different levels of abstraction, such that full configurations still stay consistent with the consultant's original configuration.

4. PL4X - CONCEPTUAL SOLUTION

Software product line engineering already provides many tools and techniques for dealing with variability and configurability of systems. Typically most product lines are adopted and maintained over a long period of time to shorten the time to market and to enhance quality. Our goals are similar, but we focus more on the early phases of an ERP system development, such as the sales phase (Obj. 1), requirements elicitation and analysis (Obj. 2) and early system design (Obj. 3).

We follow the traditional two-phase approach to product line engineering. In the first phase (domain engineering) we establish the product line by identifying its scope and explicitly documenting the variation points. During the second phase

(application engineering) we envision the automation of the sales phases and customer-specific application configuration.

4.1 Domain Engineering

The key concept to establishing our envisioned solution is a self-updating variability model. The model is linked to a software repository and ERP domain artifacts such as classes, configuration settings and table definitions. We call this enhanced software repository the *Feature Model Store*. We foresee the Feature Model Store to provide an automatic population mechanism for the variability model (see Figure 2).

4.1.1 Variability Mining and Modeling

As many customer-specific configurations have already been developed, the first step is to understand the rationale behind these adaptations, in order to make the variation points explicit. These variation points (e.g., table definitions, forms, reports, classes) are organized and structured using their unique configuration keys. These configuration keys can be compared to features and Microsoft Dynamics AX provides tools to model and manage such keys. Extracted customizations (as highlighted as step 1 in Figure 2) with their configuration keys (step 2) are mapped to the corresponding requirement description from a requirements management system (step 3), thus establishing the traceability between the problem space and the solution space. This can be done using an ALM tool like Microsoft Team Foundation Server that links the version control system with the developer's work items. In other cases this can be done by applying a naming convention where a requirement ID is used in the IDE to group artifacts, e.g. Group #27 belongs to the IDE implements requirement #27. The extracted artifacts (customizations, configurations keys and requirements descriptions) are compiled into a new unified artifact (step 4), called PL4X Feature Element. These artifacts are integrated with the variability model (similarly to how requirements and variability model are integrated in [16]-[18]) and stored in the Feature Model Store. Figure 2 highlights these key steps.

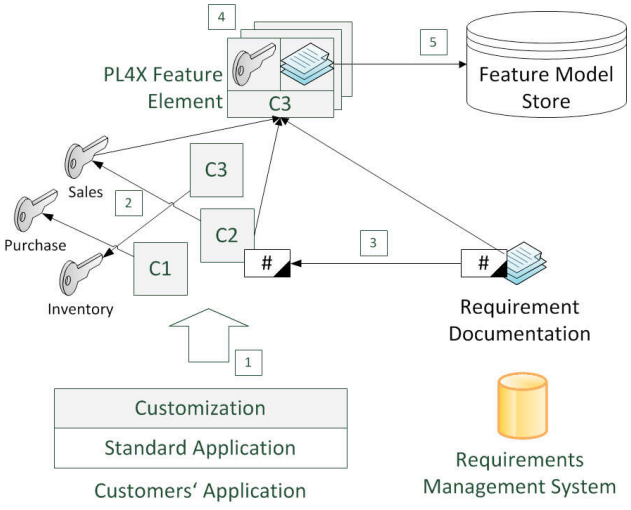

Figure 2 ***Populating the Feature Model Store***

In the ongoing research we have evaluated the feasibility of creating such a Feature Model Store. Several of the envisioned steps (1 to 3) can already be conducted by using state-of-the-art ERP tools. Figure 3, for example, shows a part of the existing configuration framework of Dynamics AX, which allows managing, activating and deactivating features. In this example all configuration keys for Logistics are activated.

We have developed an initial prototype of the feature model store based on the available Configuration Framework. The current solution would allow the manual set-up and management of the Feature Model Store. However, we envision the automation of this process as manual management is time consuming and would outrage the expected benefits.

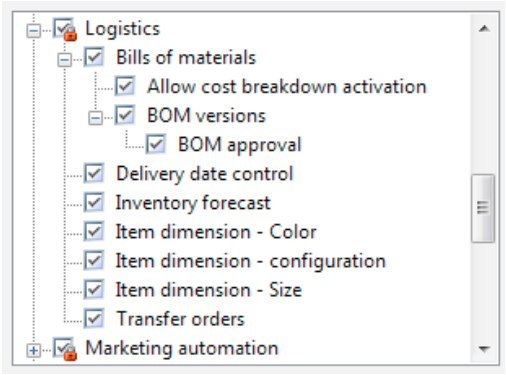

Figure 3 ***Management of configuration keys in Dynamics AX is comparable to feature modeling, but too fine-grained.***

4.1.2 Maintenance of variability models

We are currently working on a method that would allow the automated population of the Feature Model Store. We foresee a crawler mechanism, which continuously analyses existing solutions and performs the presented population steps (see Figure 2) automatically. This would ensure limited maintenance effort regarding the Feature Model Store and accurate and up to date models. As this is ongoing research work and a Feature Model Store population has not been generated so far, maintenance activities are yet not planned in detail.

However, we envision a Feature Model Store that is free of redundancy (i.e. cross-cutting concerns). Whenever multiple features require exactly the same configuration settings, these settings could be extracted into one common parent feature that gets extended by the remaining configuration settings that do not overlap. This firstly generates constraints (i.e. child-features depend on parent-features) and eases a later configuration, where constraint propagation can be used. And secondly, well-defined, non-redundant variable features may make the variability model smaller, better understandable and better maintainable. Refactoring approaches as e.g. in [17] could be used to ease such maintenance tasks for engineers.

4.2 Application Engineering

In the following we describe how the information provided by the Feature Model Store can be used in ERP projects. We therefore highlight different scenarios, which also refer to the presented research objectives.

4.2.1 Sales Scenario

The following scenario refers to Obj. 1 and discusses the envisioned support for sales activities within our approach.

The goal of the sales scenario is to convince a customer about the competences of a partner company, so that the customer chooses the company for project realization. This competitive phase bears

high risks for an ERP partner company. They might spend time and money in the construction of a sales prototype, but lose the competition to a competitor.

Currently, a role-tailored questionnaire supports the technical sales specialist to cover the most important topics relevant for building the sales prototype. The analysis of the customer responses and the configuration of the prototype are time consuming and costly steps.

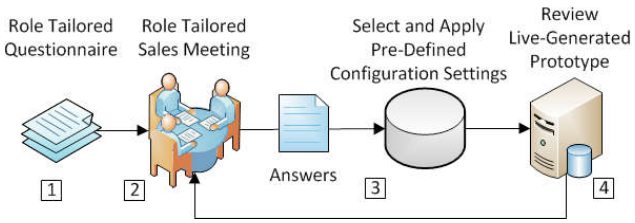

Figure 4 ***Steps of a sales scenario: from a customer-specific questionnaire to a prototype for demonstration.***

We foresee to automate this activity of generating a prototype system. We provide a questionnaire where particular questions are linked to existing configuration settings (see Figure 4). This would result in a process where the technical sales specialist still discusses the questionnaire with the customer's key users. Answers to questions would lead to a specific configuration setting. These settings are immediately applied to a live-generated prototype and discussed with the customer's key users. A first tool prototype supporting the sales scenario is presented in Section 5.

4.2.2 Analysis and Design Scenario

The following scenario refers to Obj. 2 and 3 and discusses requirements elicitation and early system design.

The goal of the analysis scenario is to gather and discuss customer requirements and to identify features fulfilling customer needs. Furthermore, a Fit/Gap analysis is conducted in order to also identify missing functionality.

The analysis is conducted at the customer's place in the form of workshops. Key stakeholders from a specific department (e.g. finance, sales) are invited to discuss their requirements. In some cases a developer who provides immediate ERP system prototypes accompanies the ERP consultant facilitating the workshop. These prototypes (e.g., user interface prototypes) support the consultants in clarifying specific issues by demonstrating system behavior and stimulating the discussion. However, in most cases there is no on-site developer and consultants have to communicate initial requirements to developers at the ERP Company, who then start the development of needed prototypes. This prolongs the requirements and analysis scenario.

Currently, most prototypes are built for demonstration purpose only; they reflect a small part of the future system and will be deleted when no longer needed. Building these prototypes is time consuming and costly and does not lead to prototypes showing full system capabilities.

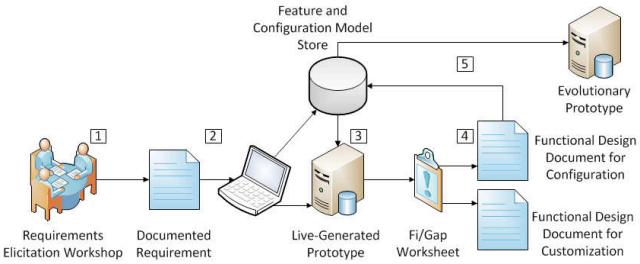

Figure 5 ***Steps of an analysis scenario: from requirements to design documents and evolutionary prototype.***

We know from our past research [5, 6] that consultants can benefit from mapping requirements to features as part of a requirements elicitation workshop. In our ongoing research we will extend the rapid prototyping mechanism as discussed in the sales scenario (see 4.2.1) and provide tool support for consultants to identify customers' requirement and to map features as stored in the Feature Model Store (see Figure 5). Selected features can immediately and automatically be deployed into a prototype, which continuously grows during the workshop. Ideally, at the end of a workshop, the prototype reflects all customer requirements that can be satisfied by using existing features. This information is also documented in the Fit/Gap worksheet, which also describes customer requirements that cannot be satisfied using existing solutions. For those requirements (i.e., gaps) the partner company has to build individual solutions.

Although this solution makes heavy use of existing features, modifications will most likely be needed to meet customer's individual needs. However, in contrast to the original prototyping approach, our solution delivers a prototype based on well-tested features, instead of quickly developed customizations. Therefore, it has the potential to be reused after the workshops.

5. PROTOTYPIC TOOL SUPPORT

We have implemented an early tool prototype called PL4X ERP Configurator (see Figure 6). The tool provides role tailored questionnaires from Dynamics Sure Step [7], a guide from Microsoft for partner companies to manage ERP projects. Every question has one or more typical answer options. Each of these options is linked to a corresponding configuration for the ERP System. An example for such a question could be: *How do you track data modifications made by users?* This question could provide the following options: "*We don't track modifications*"; "*We track modifications on customer contact information*"; or "*We track creation of new items*". Answering the question using the first option would not require any action. Option two and three would lead to different configuration settings. In our example this would result in a modified *DatabaseLog* table.

The PL4X ERP Configurator can manage multiple role and project-specific questionnaire instances and customer specific answers. Figure 6 shows how the presented questions could be discussed with the help of the configuration tool.

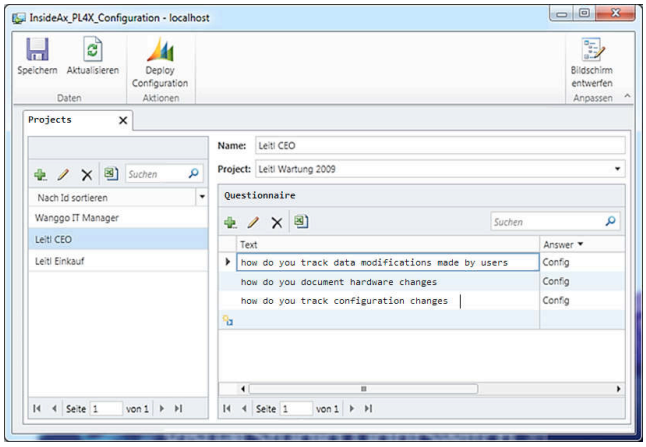

Figure 6 ***Screenshot of a PL4X ERP Configurator prototype.***

6. RELATED WORK

Our work is related to many other papers dealing with variability modeling and product line management techniques. Apart from generic approaches to variability management and traceability to different artifacts, e.g., [8, 9, 10, 11], our research objective is similar to the ones followed by specific papers in variability of ERP systems (e.g., [12, 13]).

Our research clearly shows that product line management system based on Dynamic AX is a software ecosystem, as described in [3]. The three different levels and roles of stakeholders as described by [12] are also present in our scenarios. Therefore, we are currently working towards applying similar techniques for a new ERP system. Nevertheless, our approach will have to address domain-specific requirements and handle the constraints described in section 2.

Although, theoretically, any variability modeling approach could be adopted, the stakeholders of ERP systems are used to working with dynamic questionnaire-based configuration systems, which have motivated us to follow a question-answer based approach. Therefore our approach shows some resemblance to DOPLER tools [11]. Nevertheless, the similarity is only superficial, as our models are based on Dynamic AX configuration keys and are automatically populated and self-updating, unlike DOPLER.

There also exists some work on automatically generating feature models from existing artifacts, which is similar to our feature model mining solution. For example, Weston et al. [14] uses an early aspects mining tool to find candidate cross-cutting concerns over textual requirements documents of different products, by their semantic similarity. Thus, their approach allows to automatically constructing feature models from diverse and heterogeneous requirements documents. Further, Wang et al. [15] introduce a formal model of use cases based on which application feature models are constructed after some preprocessing. These models are then adjusted and finally merged into a domain feature model semi-automatically, after all conflicts have been removed. Both approaches extract the feature models from existing requirements artifacts. In this project, we populate the Feature Model Store by mining the solution space – more concretely, the specific parameterizations and vertical solutions of specific Dynamics AX customizations.

Considering the interdisciplinary nature of ERP tools, our approach follows a step-wise refinement of the configuration, which is similar to the ideas of a staged configuration [9]. Another important aspect is the stakeholder-role-based perspective on the variability models, as illustrated by [12]. In this context, we are looking into different visualization techniques, considering the cross-cutting nature of features over different kinds of artifacts, for which we get some inspirations from [10] and [16].

7. DISCUSSION AND CONCLUSION

This paper described research challenges related to adoption of product line techniques in an ERP ecosystem from the perspective of a partner company. Such problems have not yet been sufficiently addressed by other research efforts, as most approaches consider either the perspective of the main vendor or the end-users (customers). We consider our work to provide initial contributions, but see several issues. This includes, but is not limited to the following:

Return on investment: Every 2-3 years Microsoft provides a new release of Dynamics AX, which invalidates the generated and manually refined configuration keys in the envisioned Feature Model Store. Thus, the first product customization after a new release will need to be done manually. Based on these customizations it will be possible to auto-generate a new population for the Feature Model Store. We foresee that the decrease in customization costs will clearly outweigh the introduction and maintenance costs of our approach. However, evaluation studies are needed to validate this claim.

Redundancy in mined features: The feature model mining process may populate the Feature Model Store with features that may overlap in content. Such overlaps lead to redundancy and, thus, to the problems typically associated with cross-cutting concerns. Therefore, frequent refactoring of the Feature Model Store may be required which extracts common contents of multiple features into parent-features and leaves the heterogeneous customizations as child-features. Techniques as presented in [17] may help to minimize the effort for reducing such redundancy.

Pre-configured solutions for specific industries: ERP solutions for specific domains (e.g., construction industry) are similar to some degree. So far we do not consider this issue and it would be advisable to include settings for specific industries in our conceptual solution. Pre-selection of features for a specific domain may be codified as variability constraints.

Although these open questions do exist, we consider our work to provide initial contributions. This paper identified some challenges related to tailoring ERP systems, from the perspective of a partner company, building upon a third party product platform. We discussed some limitations affecting the introduction of product line techniques in the ERP domain. Furthermore, we provided first conceptual solutions and discuss an early tool prototype. The presented PL4X approach, which tailors software product line techniques to the ERP domain by also considering domain specific constraints, is the main contribution of our work.

We expect a competitive advantage for ERP partner companies applying the PL4X approach. The novel approach strengthens the role of the sales specialists by providing tools for rapid prototype configuration. We expect a noticeable reduction of the time needed for building prototypes by using automation mechanism. Furthermore, we expect more complete and precise requirements and an even higher quality of delivered software solutions (e.g., regarding maintainability and robustness) due to the heavy reuse of existing features.

Next steps in our research include work on the conceptual solution and the development of useable prototype solutions. We plan to evaluate these prototypes with the help of sales consultants and developers at different ERP partner companies [18, 19].

ACKNOWLEDGMENTS

The research conducted was in part funded by the Austrian Research Promotion Agency (FFG, Project Nr.: 2025605).

REFERENCES

[1] Clements, P. & Northrop, L... *Software Product Lines: Practices and Patterns*, Addison Wesley, 2001

[2] Van der Linden, Frank, Schmid, K., & Eelco Rommes. *Software Product Lines in Action*, Springer, 2007

[3] Bosch, J., "From software product lines to software ecosystems". In Proceedings of the 13th International Software Product Line Conference (SPLC '09). Carnegie Mellon University, Pittsburgh, PA, USA, 111-119.

[4] Pohl K., Böckle G., Van der Linden F., Software Product Line Engineering, Foundations Principles and Techniques, Springer 2005

[5] Nöbauer M., Seyff N., Maiden N., Zachos K.; S³C: Using Service Discovery to Support Requirements Elicitation in the ERP Domain; Conference on Advanced Information System Engineering CAiSE'11, 2011

[6] Nöbauer M., Seyff N.; Conducting Research in an SME Company: A Discussion of Success Factors and Risks; Third Workshop on Leveraging Empirical Research Results for Software Business Success, 2011

[7] Project Managing Microsoft Dynamics AX, Microsoft Dynamics GP and Microsoft Dynamics NAV Implementations with Microsoft Dynamics Sure Step 2010, Microsoft Corporation, 2010

[8] Kang, K.C. and Cohen, S.G. and Hess, J.A. and Novak, W.E. and Peterson, A.S., "Feature-oriented domain analysis (FODA) feasibility study", Technical Report CMU/SEI-90-TR-021, SEI, Carnegie Mellon University, November 1990

[9] Czarnecki, K. and Helsen, S. and Eisenecker, U., "Staged configuration using feature models", Proceedings of the Third International Conference on Software Product Lines (SPLC '04), volume 3154 of Lecture Notes in Computer Science. Springer Berlin/Heidelberg, August 2004.

[10] Stoiber, R., Meier, S., and Glinz, M. „Visualizing Product Line Domain Variability by Aspect-Oriented Modeling". In *Proceedings of the Second International Workshop on Requirements Engineering Visualization (REV '07).* IEEE CS.

[11] Dhungana, D., Grünbacher, P., Rabiser, R.: The DOPLER Meta-Tool for Decision-Oriented Variability Modeling: A Multiple Case Study. Automated Software Engineering, Springer, vol. 18(1), pp. 77-114, 2011.

[12] Rabiser, R., Wolfinger, R., Grünbacher, P.: Three-level Customization of Software Products Using a Product Line Approach. Proc. of the 42nd Hawaii International Conference on System Sciences (HICSS-42), Waikoloa, Hawaii, USA, January 5-8, 2009, IEEE Computer Society, pp. 1-10.

[13] Wolfinger, R., et. al., "Supporting Runtime System Adaptation through Product Line Engineering and Plug-in Techniques". Seventh International Conference on Composition-Based Software Systems (ICCBSS 2008), Madrid, Spain, 25-29 Feb. 2008, pp. 21-30

[14] Weston N., Chitchyan R., Rashid A., A framework for constructing semantically composable feature models from natural language requirements, in Proc. of SPLC'09. ACM, 2009.

[15] Wang B., Zhang W., Zhao H., Jin Z., Mei H., A use case based approach to feature models' construction, in Proc. of RE'09. IEEE CS, 2009.

[16] Stoiber, R., M. Glinz (2009). Modeling and Managing Tacit Product Line Requirements Knowledge. *Proceedings of the 2nd International Workshop on Managing Requirements Engineering Knowledge (MaRK'09).* IEEE CS.

[17] Stoiber R., Glinz M.: Feature Unweaving: Efficient Variability Extraction and Specification for Emerging Software Product Lines. In: 4th International Workshop on Software Product Management (IWSPM 2010). IEEE CS, 2010.

[18] Stoiber R, Glinz M, Supporting Stepwise, Incremental Product Derivation in Product Line Requirements Engineering. In: 4th International Workshop on Variability Modelling of Software-Intensive Systems, Proceedings 2010

[19] Batory D., Feature Models, Grammars, and Propositional Formulas, In: Software Product Lines, 9th International Conference, SPLC 2005, Proceedings 2005