

# A Reuse-Oriented Product-Line Method for Enterprise Web Applications

Neil Mather and Samia Oussena

School of Computing and Technology  
University of West London, London, UK, W5 5RF

**Abstract.** Software product line engineering (SPLE) is a methodology for achieving systematic asset reuse in a family of software. The author of this proposal is producing a range of enterprise web portal products for Higher Education Institutions. The commonalities and variabilities of this product family suggest a SPLE approach would be beneficial. However, research indicates that full-blown, proactive SPLE is not always suited to small businesses. Efforts exist to reduce the overheads of SPLE. In this vein, this research proposes to develop a method for applying software product line engineering to enterprise web application development that makes efficient use of existing frameworks. This research falls into the domain of model-driven processes and methods for web engineering.

## 1 Introduction and Motivation

This research involves the creation of a software product line for enterprise web portals. The portals from this product line are to be deployed to Higher Education Institutes (HEIs) in various domains, such as nursing, social care, occupational therapy and teaching, for the administration of practice-based learning.<sup>1</sup>

While there are many similarities between practice-based learning in each of these domains, there are also many subtle (and not-so-subtle) differences. These differences are not just between domains, but also between HEIs. The variabilities can be in many places. There can be differences in the concepts and business processes in the domain. For example, while a nursing student may have a “mentor”, in social work this may be their “practice educator” – a similar yet different concept. Similarly, the process of sending a student on placement varies from HEI to HEI. There can also be differences in the basic functionality of a portal for each HEI – each has its own set of requirements for basic functionality and customisations such as branding and available features. In developing software systems for practice-based learning across domains and HEIs, techniques are required to account for these commonalities and variabilities. Traditional approaches to software reuse are not suited to this purpose. This research seeks to utilise SPLE to enable efficient reuse in the creation of these enterprise portals, and to investigate the use of SPLE in web applications at large.

---

<sup>1</sup> Practice-based learning is a form of higher education wherein a significant portion of a student’s education is spent training in real-world environments.

## 2 Background

SPLE is a systematic approach to achieving software reuse. It aims to minimise the overheads incurred when building a family of software products that have significant shared features, yet various differences. SPLE and its reuse-oriented methodology has many purported advantages, both financial and technical, and a number of industrial case studies document the successes it can bring [9,14]. Yet due to its relative newness and its perceived overheads it is not yet in widespread use in small- to medium-sized enterprises [8, p. 205]. Efforts exist to bring the benefits of SPLE to companies less able to absorb its large up-front analysis times [7,5,11].

SPLE is commonly split into two main processes: domain engineering and application engineering [13, p. 20]. In *domain engineering* the scope of the domain of the product family is defined, and common and variable parts across products are identified. *Variability management*, a key component of SPLE, is used to manage these commonalities and variabilities. *Feature modelling* is the most common method for variability management [3]. In *application engineering* concrete products are produced from the assets and models that have been produced in the domain engineering process. This requires the production of a *configuration specification* of the variability model, which defines the features to be included (and excluded) from the product in question. A transformation process must then take place, mapping the assets and the configuration specification into a concrete product. While frequently the goal is for this mapping to be automated, human intervention in the process is often required.

Enterprise portals are web-based systems that provide the means for researching, collaborating and manipulating data within the enterprise. They provide functionality such as targeted information provision, easy content management, inter-personnel communication and collaboration, business intelligence reporting and quick access to line-of-business data [4]. The utility of enterprise portals has infiltrated the corporate consciousness enough for several pre-fabricated, commercial portal frameworks to exist, which can be used by companies to ‘roll their own’ portal without having to start entirely from scratch. These portal frameworks are predominantly component-based architectures.

Web systems are seen as an area where the rapid response times of SPLE for new products can be beneficial. However, the high upfront costs can be off-putting to small- to medium-sized enterprises. The extractive and reactive approaches are ways around these costs [7], and similarly the use of pre-existing tools can be another time-saving measure [8]. Despite this, much prior work in integrating the SPLE paradigm with web applications and portals has involved the construction of a new, custom-built portal framework each time [12,8,2,1]. This is a missed opportunity for software reuse. This research will suggest that a process for variability management and application engineering using an *existing* portal framework will further increase the productivity gains of SPLE in the sphere of the web.

### 3 Aims and Objectives

The specific aims of this research are:

- To demonstrate that the software reuse savings of SPLE can be further increased by making use of existing portal and web frameworks.
- To build upon prior work and develop the concepts around SPLE and web systems, and to define new concepts where required.
- To develop a methodology that describes how to take an ‘off-the-shelf’ portal/web framework and apply variability management and SPLE techniques to produce applications in a product line.

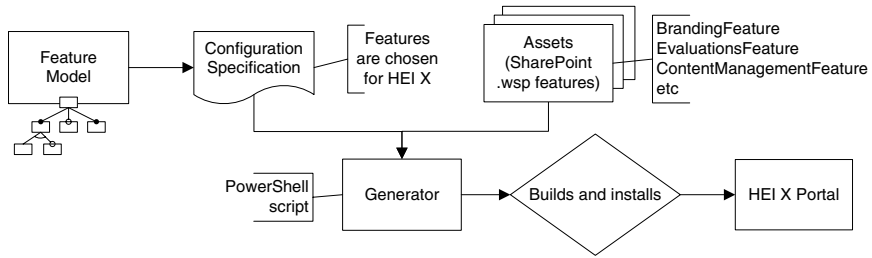
### 4 Research Methodology

This research will begin with a design science creational phase, followed by an empirical evaluative stage [10]. The first stage will create the software artifacts and methodologies, and the second stage will empirically investigate the utility of these artifacts in a real-world context.

The first stage will utilise the *Formulative-Process* research approach, and the research method will be *Concept Implementation*. These are the most predominant approaches in software engineering research [6]. These methodologies will be used to create the research artifacts of this research – newly defined concepts for the use of SPLE with web systems; a general methodology for reusing frameworks for web systems; and an instantiation of this methodology in one technology.

The product-line will be defined in a reactive fashion based on demand, as opposed to the heavy up-front analysis of the proactive approach [7]. The core and variable assets and models will evolve as new portals are created for new customers. Initial portals will go through the software product life-cycle of requirements engineering, analysis, design, implementation and testing. As the portal product-line is produced, the toolkit for portal generation will be developed. This artifact will take the form of a generator that, given a configuration specification (derived from a variability model such as a feature model), will produce a generator script. This script will compile and install the required assets in the portal framework to produce an individual product. This toolkit will be used to inform the definition of the generalised methodology for reusing portal frameworks in an SPLE approach.

The initial evaluation of the method will be based on the *proof by implementation* research method. The construction of a fully-operational product line from which applications can be produced and sold will be taken as partial validation of the method. It is important, however, to more rigorously validate the method in comparison to alternative methods. This can be performed quantitatively by analysing the reduction in developer-written lines of code by the method. Further software reuse metrics can also be employed. In addition to this, empirical evaluation via case study will be performed in the second stage of the research.



**Fig. 1.** Process for generating a portal from a configuration specification and pre-built assets. Selected assets are installed by the Generator. Future iterations will allow for variability within individual assets, for example through compilation-time bindings in source code, or run-time parameterisation.

## 5 Research Plan

Work to date has involved the production of conceptual models, business process models, and domain glossaries of the social work and nursing domains of practice-based learning. Research into portal frameworks has been performed, as has research into product line and variability management techniques. Initial requirements engineering has been performed with one HEI in the social work domain. Use cases have been created and a prototype portal has been developed using C#, ASP.NET, and Microsoft SharePoint. This portal is currently being tested by the HEI. A prototype version of the toolkit for portal generation has been produced, as shown in Figure 1.

Going forward into the 2nd year, the reactive SPLE approach will be used as portals are produced for other HEIs. This will define the practice-based learning portal product-line as variability is incorporated into the existing models and assets. As this process occurs, concepts for web systems product-lines will be defined, with a focus on reuse of the portal framework infrastructure and components. The toolset with which to automate the generation of new web applications from product-family assets will be enhanced. The concepts of this toolset will be generalised to describe a methodology applicable to portal frameworks and web systems in general. In the 3rd year of research, empirical, evaluative analysis of the utility of the methodology and its instantiation will be made via case study. This will be achieved at the company as more portals are rolled out to customers.

## 6 Contributions to Web Engineering

The main contributions of this research will be:

- Definition of concepts related to SPLE and enterprise web systems. This will include variability management techniques for enterprise ontologies and business processes, and investigations into aspect-oriented programming for web-related software product lines.

- A general methodology with which existing portal/web frameworks can be used in the software product line engineering paradigm.
- An instantiation of the methodology – i.e. a toolkit that can be used to generate enterprise web applications incorporating the reuse of one existing framework technology.
- An empirical, industrial case study of the methodology in practice.

## References

1. Balzerani, L., Di Ruscio, D., Pierantonio, A., De Angelis, G.: A product line architecture for web applications. In: *Proceedings of the 2005 ACM Symposium on Applied Computing*, pp. 1689–1693. ACM (2005)
2. Capilla, R., Dueñas, J.C.: Light-weight product-lines for evolution and maintenance of Web sites. In: *Proceedings of the Seventh European Conference on Software Maintenance and Reengineering*, pp. 53–62. IEEE Computer Society (2003)
3. Chen, L., Babar, M.A.: A Status Report on the Evaluation of Variability Management Approaches. In: *13th International Conference on Evaluation and Assessment in Software Engineering (EASE)*. BCS (2009)
4. Dias, C.: Corporate portals: a literature review of a new concept in Information Management. *International Journal of Information Management* 21(4), 269–287 (2001)
5. Ghanam, Y., Maurer, F.: Extreme Product Line Engineering: Managing Variability and Traceability via Executable Specifications. In: *Proceedings of the 2009 Agile Conference*, pp. 41–48. IEEE Computer Society (August 2009)
6. Glass, R.: Research in software engineering: an analysis of the literature. *Information and Software Technology* 44(8), 491–506 (2002)
7. Krueger, C.W.: Easing the transition to software mass customization. In: *Software Product-Family Engineering*, pp. 178–184 (2002)
8. Laguna, M., González-Baixaui, B., Hernández, C.: Product Line Development of Web Systems with Conventional Tools. In: *Proceedings of the 9th International Conference on Web Engineering*, pp. 205–212. Springer, Heidelberg (2009)
9. Van Der Linden, F., Schmid, K., Rommes, E.: *Software product lines in action: the best industrial practice in product line engineering*. Springer, Heidelberg (2007)
10. March, S.T., Smith, G.F.: Design and natural science research on information technology. *Decision Support Systems* 15(4), 251–266 (1995)
11. McGregor, J.D.: Agile Software Product Lines - A Working Session. In: *Proceedings of the 2008 12th International Software Product Line Conference*, vol. 7. IEEE Computer Society (September 2008)
12. Pettersson, U., Jarzabek, S.: Industrial experience with building a web portal product line using a lightweight, reactive approach. In: *Proceedings of the 10th European Software Engineering Conference*, vol. 30, pp. 326–335. ACM (September 2005)
13. Pohl, K., Böckle, G., Van Der Linden, F.: *Software Product Line Engineering: Foundations, Principles and Techniques*. Springer, Heidelberg (2005)
14. Software Engineering Institute. *Catalog of Software Product Lines* (2010)