Transitioning Legacy Assets to a Product Line Architecture

Joachim Bayer*, Jean-François Girard*, Martin Würthner*, Jean-Marc DeBaud†, Martin Apel‡

* Fraunhofer Institute for Experimental Software Engineering (IESE), Sauerwiesen 6, D-67661 Kaiserslautern, Germany
  +49 (0) 6301 707 251  {bayer, girard, wuerthne}@iese.fhg.de

† Lucent PaceLine Technologies, LLC,
  263 Shuman Boulevard, Naperville, IL 60566, USA
  +1 (630) 224 0383  debaud@research.bell-labs.com

‡ Tecmath GmbH & Co. KG
  Sauerwiesen 2, D-67661 Kaiserslautern, Germany
  +49 (0) 6301 606 80  apel@tecmath.de

Abstract. A successful software system evolves over time, but this evolution often occurs in an ad-hoc fashion. One approach to structure system evolution is the concept of software product lines where a core architecture supports a variety of application contexts. However, in practice, the high cost and high risks of redevelopment as well as the substantial investments made to develop the existing systems most often mandate significant leverage of the legacy assets. Yet, there is little guidance in the literature on how to transition legacy assets into a product line set-up.

In this paper, we present RE-PLACE, an approach developed to support the transition of existing software assets towards a product line architecture while taking into account anticipated new system variants. We illustrate this approach with its application in an industrial setting.

Keywords: software product line, architecture recovery, reengineering, reuse, domain-specific software architecture

1 Introduction

1.1 Problem
One of the principal characteristics of a successful software product is that, over time, it will be reused and adapted for purposes that become increasingly different from the product’s original purpose. At the same time, customer, project, and organizational pressures make it very difficult to coherently manage the resulting multiple development and maintenance threads.

Apart from unnecessary duplication of effort, some of the possible consequences are structure degradation, un-managed diversity, and little conceptual integrity among...
the system variants. These adverse consequences often imply additional evolution time and cost. This is a widespread situation that affects many software systems today.

1.2 Approach and Context

One possible way of addressing the problems mentioned in the previous section is the product line concept. A product line is a set of systems sharing core features while varying in others. It may be operationalized via a domain-specific software architecture, which, using a flexible and customizable design, embodies a platform from which single systems can be derived.

Although most of the product line approaches suggest integrating existing assets into the product line reference architecture, there is little guidance on how to achieve this. Our strategy was to develop an approach to help organizations transition their existing assets to a new, more modern design. This design should support the evolution of existing applications and accommodate future applications as far as possible. The result is RE-PLACE, an approach detailed and illustrated in this paper.

RE-PLACE stands for Reengineering-Enabled Product Line Architecture Creation and Evolution. RE-PLACE was developed during the RAMSIS kernel redesign project to communicate the approach we were taking and to allow its application in different contexts. RAMSIS is a large, very successful human modeling system. Its kernel has evolved as a monolithic Fortran system over the past 10 years and has grown to a size where all maintenance tasks and new developments need considerable time. Therefore, it is not very well-suited for the anticipated new uses in various application contexts, which makes the kernel redesign necessary.

1.3 Related Work

To develop RE-PLACE, we explored both emerging product line concepts and techniques, and the existing reengineering ideas and methods: product line concepts and techniques because they provide us with the basis for thinking in terms of a family of systems, and reengineering ideas and methods because they allow us to leverage existing assets.

There are a number of domain engineering approaches proposed such as Organizational Domain Modeling (ODM) [21], Model-Based Software Engineering (MBSE) [15], the Domain-Specific Software Architecture (DSSA) program [22], and Synthesis [20].

Although the need for reusing existing assets in domain engineering has been recognized, there has been little published work in the past that tried to leverage existing assets by integrating them into a product line architecture. Existing work [6, 23] includes process overviews of possible mechanisms and entry points for integrating reengineering activities in product line development.

Reengineering has a rich tradition concerning the recovery of software assets and their integration within a new environment. One key reengineering activity in transitioning a system is the recovery of its architecture. Many approaches to architecture recovery have been proposed [7, 9, 10]. Domain-Augmented Re-Engineering (DARE) [5] integrates the domain dimension, showing how domain model construction and reverse