December 2003

Domain-Specific Component Design through Product Platforms

Marcus Rothenberger
University of Wisconsin, Milwaukee

Hemant Jain
University of Wisconsin, Milwaukee

DOMAIN-SPECIFIC COMPONENT DESIGN THROUGH PRODUCT PLATFORMS

Marcus Rothenberger
School of Business Administration
University of Wisconsin, Milwaukee
rothenb@uwm.edu

Hemant Jain
School of Business Administration
University of Wisconsin, Milwaukee
jain@uwm.edu

Abstract

During recent years, traditional custom software development has evolved into a variety of software adoption philosophies driven by advances in technology and an increasing demand for software products. One emerging philosophy is the use of component markets that support the notion of inter-organizational reuse. Currently, available components are mostly limited to infrastructure items, such as user interfaces, printing components, and data access modules. This is because the functionality of infrastructure components is highly standardized and can be clearly defined for retrieval through search engines. However, without the ability to build the core of an application around reusable software components, the true value of component-based software development cannot be obtained. Thus, in order to move this adoption philosophy forward, domain-driven components must be available for various domains.

This research draws from the lessons learned from product platforms in the manufacturing world to investigate how domain-specific components can be designed for successful black-box reuse. Each component platform must unify the functional requirements of a common user base. Based on platform-development in manufacturing, the design of a platform must be guided by the customer demand. In terms of component-platform development, that means that the projected market for the component must drive its design. Market demand for derivative software components must equally be anticipated and a list of functionalities that change across derivatives must be identified. All the functionality that is not anticipated to be changing across these dimensions may be incorporated into the component platform. Other functionality may be incorporated into separate "plug-ins" that allow the component user to customize according to specific requirements.

We are going to validate this approach by applying it to a software product of a software developer in the financial services industry. We will demonstrate how to meet the requirements of multiple distinct target groups with the help of a product platform-driven design. We are going to compare the platform-based design with the traditional design of the product to assess design quality and user requirements met. This step will involve the use of expert panels of application designers of the subject organization.