Model Driven Architecture Development Approach for Pervasive Computing

Kai Hemme-Unger
DaimlerChrysler AG
P.O. Box 23 60
89013 Ulm
+49-(0)7 31-5 05-24 06
kai.hemme-unger@web.de

Thomas Flor
DaimlerChrysler AG
P.O. Box 23 60
89013 Ulm
+49-(0)7 31-5 05-28 59
thomas.flor@
daimlerchrysler.com

Gabriel Vögler
DaimlerChrysler AG
P.O. Box 23 60
89013 Ulm
+49-(0)7 31-5 05-48 70
gabriel.voegler@
daimlerchrysler.com

ABSTRACT
This paper describes a model-based development approach for pervasive computing applications. The key concept introduced is the use of the Model Driven Architecture (MDA) for development of system families.

Categories and Subject Descriptors
D.2.13 [**Software Engineering**]: Reusable Software - *domain engineering*; D.2.9 [**Software Engineering**]: Management – *productivity, life cycle.*

General Terms
Design

Keywords
Component Composition, Generic Modeling, MDA, Model Driven Architecture

1. INTRODUCTION
Pervasive computing applications are the link between existing business services and client devices with differing characteristics and features. The notion of having data and processes accessible anywhere and anytime enforces the ability to quickly and efficiently adapt existing applications to new devices and applications. We consider a model-based development approach to be most suitable for this purpose.

2. DEVELOPMENT CONCEPT

2.1. Requirements
The vision of pervasive requires new approaches for software engineering. In particular, an important economic requirement is the conservation of value of existing IT investment as well as technical requirements such as scalability and adaptability. In the respective project, platform independence is a requirement of subordinate priority.

2.2. Architectural Design
In order to conserve existing IT investment, we add an overlaying Pervasive Service Layer to the existing architectural layers. It serves as a component-based run-time environment to facilitate future changes. Our goal is to devise a model-integrated application development approach to ensure greatest possible efficiency.

2.3. Model Driven Architecture
The Object Management Group's (OMG) Model Driven Architecture (MDA) emphasizes the importance of modeling for software development. The dedicated objective of the MDA is to support platform-independent application development. Therefore, models of two different abstraction levels are created, more specifically, a distinction is made between Platform Independent Models (PIM) and Platform Specific Models (PSM).

The MDA promises an increased productivity as well as modeling quality. This is reached by applying transformation rules to models in order to eventually generate code. Thus, the programmer can focus entirely on the design of the business logic without paying attention to the complexity of the implementation [1].

A disadvantage, however, of MDA is the currently missing technical support for the development of system families. Therefore, it should be possible to define rules of how to check consistency and validity of system models at design-time already. This functionality might, in the future, be provided by an extension to the Object Constraint Language (OCL).

We employ a widely used standard UML tool for modeling the necessary components. We consider the UML profile EDOC most suitable for our requirements [2].

Since the Model Driven Architecture was designed mainly to develop discrete systems, we suggest using MDA only for the development of components, and not for business processes. Instead, the actual application is modeled using a development environment that supports the modeling of system families. For this purpose, the Generic Modeling Environment (GME) which was developed at Vanderbilt University provides the most comprehensive functionality [3].

2.4. Four-Layer Modeling Architecture
We suggest modeling pervasive systems with a model architecture consisting of four layers [4].

The tool's internal modeling profile (e.g. UML) generates the meta-model of the pervasive business applications. In the terms of MDA, this model is the PIM of the pervasive business system family. It abstractly describes standard components of a pervasive computing architecture as well as its interactions.

OOPSLA'03, October 26–30, 2003, Anaheim, California, USA.
ACM 1-58113-751-6/03/0010.

According to the MDA terminology, the PSM is derived from the PIM by adding platform-specific attributes to the standard components. The attributes are described separately from the user's and the programmer's points of view.

In the last step, the functionality of the platform-specific components have to be put into sequence. For the sake of simplicity, in this step, the modeler is only aware of the user's view on the functionality of the available components.

2.5. Development Process

The combination of MDA and generic modeling results in a development process whose phases are depicted in Figure 1.

Because the development of the components may proceed simultaneously with the application modeling, the generic modeling tool should be capable of importing the component models. Unfortunately, we are aware of no generic modeling tool at the moment that supports the standard file format for model exchange (XMI). It is conceivable to accomplish the integration with a tool that is able to convert the file formats

3. APPLICATION WORKFLOW

Pervasive Business Applications offer services that the user can accomplish by offer and acceptance. In order to receive an offer, the user launches a request and proves his authorization. When authorized, he receives an offer for the requested service according to his connection bandwidth, the limitations of his device and his personal settings.

In order to provide the user with a profound basis for his decision of whether to accept the offer or not, he receives information about the expected online time and the estimated number of required interactions for the service. The application obtains this data from the records of former transactions.

GME features the export of models which will then be interpreted. The runtime environment extracts a series of activities from the exported file. The data retrieval operations trigger events that are handled by the appropriate component.

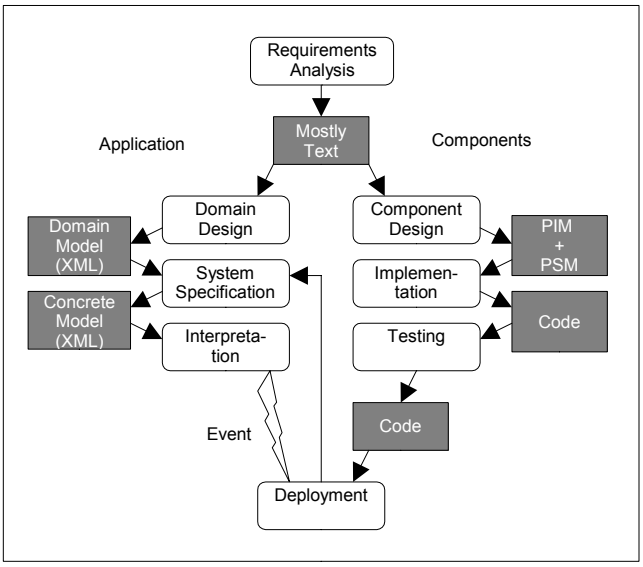

Figure 1. Life cycle

4. REFERENCES

[1] Warmer, J., Kleppe, A. and Bast, W. MDA Explained, 2003.

[2] UML Profile for Enterprise Distributed Object Computing Specification. http://www.omg.org/docs/ptc/02-02-05.pdf

[3] Lu, Y. Comparison of Meta-modeling Techniques and Tools, McGill University, 2003.

[4] Nordstrom, G. and Ledeczi, A. Formalizing the Specification of Graphical Modeling Languages. Vanderbilt University, 2000.