

# Composing Adaptive Learning Systems

Dirk Frosch-Wilke

University of Applied Sciences Kiel, Sokratesplatz 2, D-24149 Kiel, Germany

dirk.frosch-wilke@fh-kiel.de

Salvador Sánchez-Alonso

University of Alcalá, Ctra. Barcelona Km. 33.600, 28871 – Madrid, Spain

salvador.sanchez@uah.es

## Abstract

*Adaptive learning systems are recognized as one of the most interesting research topics in intelligent learning management systems. Taking as a point of departure the practices and experience in compositional adaptation of software in the computing field, we point out in this paper a number of important issues regarding adaptive learning systems, with an emphasis on the aspect of adaptation of learning systems in mobile environments and quality assurance of adaptation.*

## 1. Introduction

Adaptive learning systems (ALS) have attracted considerable attention due to their potential to provide personalized applications and services coordinated with the individual needs of learners or groups of learners.

Also in the computing field, the interest in adaptive computing systems has increased dramatically in the past few years. Now, a variety of techniques allow software to adapt dynamically to its environment, the main drivers of their appearance being the emergence of ubiquitous computing and the growing demand for autonomic computing [7]. In general terms, two approaches to implement software adaptation can be cited: *parameter adaptation* and *compositional adaptation*. *Parameter adaptation* consists of modifying those program variables that determine software behavior. *Compositional adaptation* can be defined as the dynamic recomposition of the software during execution and allows an application to adapt new algorithms for addressing concerns unforeseen during development.

When we interpret the learner model, knowledge domain, learning object repositories, learners' progressing in a learning situation, infrastructure, and

instructional model as the main parts of the current environment of an ALS during execution, the similarities between adaptation of learning systems and compositional adaptation of software become evident. In this paper, we will therefore interlink these two subjects with the aim to transfer insights in the computing field to the research topic of ALS.

## 2. Compositional adaptation

The key enabling technologies for reconfigurable software design are: Separation of concerns, computational reflection, and component design.

*Separation of concerns* refers to the ability to identify, encapsulate and manipulate those parts of software that are relevant to a particular concern. Concerns are the primary motivation for organizing and decomposing software into manageable and comprehensible parts.

*Computational reflection* enables a program to access to its internal structure and behaviour and also to programmatically manipulate that structure, thereby modifying its behaviour. It encompasses two activities: introspection (to let an application observe its own behaviour), and intercession (to let an application act on those observations and modify its own behaviour). Metaobject protocols are key techniques here [7].

*Component-based design* supports two types of composition. Static composition denotes the arrangement of software components at compile time, whereas dynamic composition allows adding, removing or reconfiguring components at runtime.

## 3. Compositional adaptation in e-learning

By using the main technologies of adaptive computing systems as taxonomy for current activities in the ALSs field, we will identify open problems and promising research directions for e-learning contexts.

### 3.1. Separation of Concerns

In the context of adaptive computing systems, the separation of an application's functional behaviour and crosscutting concerns (such as quality of service, fault tolerance, and security) are an important issue.

Although the principle of "Separation of Concerns" has partially been adopted for the development of learning systems - an approach published in [3] shows a separation of content, learner and narrative models - there still exist some crosscutting concerns that have not been widely addressed yet. Some of them should be discussed now.

*Maintenance:* One problem that arises during the design of learning objects (LOs) is that some material could conceptually belong to several LOs, making of it a crosscutting concern. In the case of duplication of such material and its metadata, the maintenance and updates are significantly complicated. In [8], the most widely used approach for separation of concerns in computing science (Aspect-oriented programming, AOP), has been proposed as a solution to the maintenance problem. In the computing field, AOP is further used to weave code fragments (aspects) that implement general crosscutting concerns into an application dynamically. In the context of ALSs this would allow to dynamically integrate material in a LO during course session. This subject has not yet been studied for ALSs.

*Assurance:* As for recomposable software design, a development paradigm is required that supports the automated checking of an ALS's 'correctness' after dynamical adaptation. The concept of correctness has two dimensions: a) consistency and quality of an adapted course, and b) quality of adaptation with respect to the current learning environment and infrastructure (such as device capabilities and communication conditions). Parameters of the first dimension of correctness are e.g. learning objectives, learner's level of knowledge, learning progress, and instructional design. In the computing field, developers can use high-level contracts [1] and invariants to monitor system correctness before, during, and after adaptation. In the field of learning objects, the concept 'design by contract' has been successfully introduced for the specification of objects responsibility [11] and extended to other participants in a learning environment [10], which could be the equivalents of the components in the computing field context. Regarding the possibility to allow automated checking of ALS's correctness, these approaches have to be extended by new contract levels like synchronization and quality-of-service parameters.

### 3.2. Computational reflection

Computational reflection will become an important issue for ALSs in the future. The advances in wireless technologies and the increasing availability of high-bandwidth telecommunication networks in recent years will increase the demand for "mobile e-learning", i.e. the use of e-learning systems on mobile devices.

Traditional e-learning systems did not take mobile user characteristics into consideration [5]. However, it seems obvious that mobile learners' requirements concerning content volume and learning environment, such as hot spot, hotel etc. are quite different from desktop learners' requirements [2]. A learning system which will operate in a multiple platform environment should be capable of reasoning about device and connectivity aspects, e.g. media support types, display capabilities, device memory or network reliability.

Thus, providing reflective services in a "mobile e-learning" environment is a critical task. These services can be either implemented in the learning system or provided by a middleware platform. In any case, design solutions for learning systems in multi-platform environments must enable "systematic" introspection and intersection of the base-level objects of the system.

Goh and Kinshuk [5] have presented an adaptation framework for "mobile learning systems" which addresses some of the above-mentioned aspects by defining concerns that they call (sub-)dimensions. However, introspection and intersection were not addressed in this work. The integration of generic object and relationship support into an ALS to enable computational reflection is a future research work which consists of two steps: a) detailed definition of the base level objects and b) definition of the self-representation (metalevel layer) of the base level objects, where we can partially use common parts of SCORM (like the Content Aggregation Model).

"Traditional" metaobject protocol implementations for adaptive computing systems are aimed at loading and unloading adaptive code by the system itself or by a developer. In the e-learning context, altering and tailoring of the learning system by the user will become a necessary feature in the near future. The learning management system (LMS) can be used in this manner as one tool in a greater *holistic learning environment* which provides the end-users with the flexibility required to move along various paths according to their learning needs and styles [12]. However, this tailoring should be done in a controlled way which does not require any programming. Greater end-user flexibility can also be achieved with the help of an extended version of the metaobject protocol as demonstrated in [6] for a payment system.

### 3.3. Component-based design

Most LMSs are composed of components for specific services which are either foundational parts of the system or extension components (plug-ins). Even though many types of learning systems are available, some key components remain common in most of them, namely user model, expert model, pedagogical model, knowledge model, and communication model.

In the context of adaptation, the reusability of some of these components is an important feature, because this is a precondition for static or dynamic composition of a learning system. Some learning models, like the learning object approach [4], are based on the object orientation paradigm and allow, to some degree, reusability of learning objects in different learning contexts and for different user models. However, in general the reuse of components in e-learning environments is still not obvious and remains difficult [9]. Some of the reasons for the low degree of reusability are: poor description of the component interfaces, instructional design implementation tangled in code, blurred separation between functional and non-functional aspects (those needed by the platform, for instance), requirement to know different languages, or suffering platform changes, among others.

Using experiences from the software engineering field we can propose (at least) two solution directions:

a) Modelling systems using abstract notations (thus creating a Platform Independent Model) and projecting it by generating the code and parameters for different platforms (this Model Driven Architecture approach has been applied to LMSs by [9]);

b) Complete modularization of the system as it is done in AOP, and consolidation of the modules (aspects) by using weaver functionality.

### 4. Conclusion

Though the adaptation of learning systems is an important research topic, many problems are still unresolved. We have presented similarities between adaptive LMSs and compositional adaptation in computer science. Using the main technologies for compositional adaptation as a taxonomy for the classification and identification of open research questions in e-learning contexts, we could concurrently derive new promising fields of study by mapping experiences and solutions from computer science onto the e-learning domain.

In our future work we will integrate AOP aspects in the concept of learning objects to improve their

maintenance and reusability. Further we are going to extend the 'design by contract' approach to implement contract management issues for LMSs to allow automated validation of adapted courses.

### Acknowledgements

Work funded by the "Plan Regional de Investigación Científica e Innovación Tecnológica" (IV PRICIT), code CAM-UAH2005/046, and supported by project UAH-PI2005-070 and the LUISA project FP6-027149.

### References

- [1] Beugnard, A.; Jézéquel, J.M.; Plouzeau, N., & Watkins, D. (1999) 'Making Components Contract Aware', *Computer* 32(7), 38-45.
- [2] Chen, Y.S., Kao, T.C., Sheu, J.P. & Chiang, C.Y. (2002) 'A Mobile Scaffolding-Aid-Based Bird-Watching Learning System' in M. Milrad, H.U. Hoppe and Kinshuk (Eds.), *IEEE International Workshop on Wireless and Mobile Technologies in Education* (pp 15-22).
- [3] Conlan, O., Wade, V., Bruen, C., & Gargan, M. (2002) Multi-Model, Metadata Driven Approach to Adaptive Hypermedia Services for Personalized eLearning. *Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, pp 100-111.
- [4] Frosch-Wilke, D. (2004) 'An Extended and Adaptable Information Model for Learning Objects'. *Proc. of the 4th IEEE International Conference on Advanced Learning Technologies*, pp 166-170.
- [5] Goh T. & Kinshuk (2004) 'An adaptation framework for Web based learning system' In *Knowledge-based Intelligent Information and Engineering Systems*, LNCS (Vol. 3213), Berlin: Springer.
- [6] Lindeberg, O., Eriksson, J., & Dittrich, Y. (2002) 'Using Metaobject Protocol to Implement Tailoring; Possibilities and Problems', *Proc. of the 6th World Conference on Integrated Design & Process Technology*.
- [7] McKinley, P., Sadjadi, S., Kasten, E., Cheng, B. (2004) 'Composing Adaptive Software', *Computer* 37(7), 56-64.
- [8] Pankratius, V. (2005) 'Aspect-Oriented Learning Objects', *Proc. of the IASTED International Conference on Web-based Education (WBE 2005)*.
- [9] Renaux E., Caron, P.A. & Le Pallec, X. (2005) 'Learning Management System component-based design: a model driven approach', *MCETECH 2005 conference*, 117-125.
- [10] Sánchez-Alonso, S., Sicilia, M.A (2004) 'Relationships and commitments in learning object metadata'. *Proc. of the 5th Int. Conf. on Information Technology Based Higher Education and Training*.
- [11] Sicilia, M.A., & Sánchez-Alonso, S. (2003) 'On the concept of 'Learning Object Design by Contract'', *WSEAS Transactions on Systems* 2(3), 612-617.
- [12] Siemens, G. (2004) 'Learning management systems: the wrong place to start learning. *ElearnSpace Weblog*. Retrieved from: <http://www.elearnSpace.org/Articles/>