PERSONALISATION IN E-LEARNING: AN APPROACH BASED ON SERVICES

George D. Magoulas
Birkbeck College, University of London
Malet street, London WC1E 7HX

Dionisios N. Dimakopoulos
London Knowledge Lab, University of London
23-29 Emerald Street, London WC1N 3QS

ABSTRACT
This paper adopts a service-oriented approach as a common framework to personalise the interaction in integrated e-learning environments. This approach facilitates the integration of commercial, in-house and open source components and applications within learning organisations and regional federations by agreeing upon common service definitions, behaviours, data and user models, and protocols. Among a variety of challenging issues in this context, the paper focuses on the identification and definition of fundamental personalisation services for integrated e-learning environments and on data models for user profiling to support these personalisation services.

KEYWORDS
E-Learning, Service-oriented Architectures, Personalised Services, User Modelling, Adaptive Web Systems.

1. INTRODUCTION
Nowadays, the majority of e-learning environments rely on management systems that support or supplement teaching and learning such as Oracle’s Think.com, Learn Direct (www.learndirect.co.uk), Blackboard (www.blackboard.com), Xtensis (www.xtensis.co.uk), Knowledge Pool (www.knowledgepool.com), Mindleaders (mindleaders.com), COSE (www.staffs.ac.uk/COSE/), Colloquia (www.colloquia.net), Lotus Learning Space (www-136.ibm.com/developerworks/lotus/products/elearning/), RDN’s Virtual Training Suite (www.vts.rdn.ac.uk) and WebCT (www.webct.com). These systems have influenced practical Web-based education and when used within a constructivist framework of learning they can actively engage the learner in the interpretation of the content and the reflection on their interpretations. Nevertheless, learners have heterogeneous backgrounds and differ in traits such as skills, aptitudes and preferences for processing information, constructing meaning from information, and applying it to real-world situations.

Personalisation has been proposed in e-learning systems as a promising approach to manage the complexity of the learning experience and accommodate learner’s needs in order to maximise the effectiveness of e-learning (Brusilovsky, 2001; Magoulas et al., 2003). Personalisation technologies are defined as approaches to adapt educational content, presentation, navigation support so that they match the unique and specific needs, characteristics and preferences of each learner or a community of learners. The spectrum of personalisation is very wide, ranging from adaptivity, i.e. automatic, system-controlled personalisation, to adaptability, i.e. user-guided personalisation (Papanikolaou et al. 2003).

This paper focuses on the adaptivity end of the spectrum. First attempts to support adaptivity in e-learning environments have been instantiated in the framework of the so-called adaptive educational hypermedia systems, which appeared in the early 90s. Research efforts in this area usually employ custom made architectures with only a small number of exceptions; see for example AHA! (De Bra, 1996; De Bra and Calvi, 1998) and KnowledgeTree (Brusilovsky, 2004). This paper proposes a service-oriented approach,
(Manes, 2003), as a general framework for the development of modular and flexible personalised systems, where components can be added, removed or replaced more easily than in traditional models of adaptive hypermedia systems, and where new applications or systems can be composed from collections of available services. This approach is different from integrating directly at the user interface level (e.g. by using portals) or at the data level (e.g. by creating large datasets or data warehouses) and may enable faster deployment of personalisation technologies as long as the needs of new components are compatible with the existing component interfaces.

Among a variety of challenges that should be addressed in this context, the paper defines what basic components are needed in a service based architecture for e-learning, and identifies what services components should offer to support adaptivity.

### 2. A SERVICE-BASED ARCHITECTURE FOR E-LEARNING

A service-based architecture (see Figure 1) can provide personalisation on the basis of well defined service behaviours and interfaces and allow various open specifications, open source toolkits and standards to be used in implementing the services. From the functional definition and scope of a specific service an abstract model of behaviour and data can be developed, which describe the expected behaviour of a realisation of this service and the data model (e.g. using XML) it deals with or exchanges. A service can be realised in a number of ways, such as a Web service (e.g. using WSDL) and Application Programming Interfaces for particular programming languages. In this approach components can consist of objects, services and processes that are related to each other, e.g. a component can organise the operation of other components. A service can be used to connect one component with another, or as a method applied on an object.

Personalisation emerges through the aggregation of a set of services that implement personalised functionalities. It can be materialised by creating, managing and storing “personal views” or relationships between information from a diverse set of existing applications. These can be tailored to the needs of individual users by combining components (which will provide the necessary functionality) and assembling services from a set of components to reduce implementation cost (see Figure 1). For example, new types of “personal” learning spaces can be composed, multiple user interfaces or portals, tailored to specific users or tasks, can be produced in this way. This of course requires a framework for the user interface or portal so that to manage the communication between services, support navigation and content presentation to each user. The user interface is supported by Personalisation Services as will be discussed next.

![Figure 1. An architecture for personalization services](image-url)
2.1 Identifying Personalisation Services for e-learning

The Services layer creates the mapping between applications, systems and data, and the service-oriented model: (i) high-level services include services, processes and objects/data structures that are shared across applications, aggregate low level services functionality, manage user/application data, define processes, control objects/services etc. For example, a personalisation service can translate application functionality into user interface features. This may depend on the state of an application and/or learner activity. The personalisation service may behave differently when the learner is not authenticated. Certain personalisation services may be offered to the learner only when an application is at a particular state or when previously called services have performed certain actions. Low level services, such as an authentication service, can be considered as general purpose; they do not rely on other services, and are standardised across all applications. For example, these may include services for object/data registration, communication etc. A set of fundamental services that can be used in e-learning applications is shown in Table 1.

Table 1. Fundamental personalisation services for e-learning

<table>
<thead>
<tr>
<th>Service</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rating</td>
<td>Creates and manages annotations for learning objects. Support for the use of secondary metadata (user ratings and text annotations) for learning resources.</td>
</tr>
<tr>
<td>Retrieval</td>
<td>Personalised information seeking: browsing though the learning objects based on taxonomies descriptions; augmented keyword-based by exploiting metadata properties of the objects according to learner goals, learner preferences, cognitive/learning style etc.</td>
</tr>
<tr>
<td>View</td>
<td>Generates personalised views over learning objects and schemata. Formats educational content for delivery to the interface, including style of organisation and presentation of lesson content and search results depending on the learner model, and the changes detected in learner behaviour.</td>
</tr>
<tr>
<td>Query</td>
<td>Provides query facilities over structured and semantic descriptions of learning objects. Narrowing down or broadening the queries depending on interests, and previous search history, information seeking strategies of learners and learner behaviour.</td>
</tr>
<tr>
<td>Recommendation</td>
<td>Allows e-learning applications that use this service to recommend educational content/learning objects based on application-specific learner history and behaviour, and metadata descriptions.</td>
</tr>
<tr>
<td>Navigation Support</td>
<td>Supports navigation though the information space of the various e-learning applications based on learner model attributes and learner long/short term behaviour.</td>
</tr>
<tr>
<td>Resource Management</td>
<td>Supports the access and management of educational resources. Allows applications that use this service to automatically determine content about appropriate search terms and the structure of metadata records that will be returned to them. It also supports the retrieval, description, and organisations of educational resources.</td>
</tr>
<tr>
<td>Learner Behaviour</td>
<td>Models the learner’s general preferences so that decisions can be made for the presentation, layout and content of the pages he/she visits. Supports the mapping of digital objects and activities against specific competencies. Allows applications, such as a portal, to automatically configure themselves for particular user(s) and to prevent users from having to enter their preferences into multiple application interfaces, such as portal, library, learning environment etc.</td>
</tr>
<tr>
<td>Learner Activity Management</td>
<td>Manages the learner activity when accessing the various applications and learning resources. It can be used to initialise services with learner preferences information when this is required by an e-learning application.</td>
</tr>
<tr>
<td>Tracking/Change Detection</td>
<td>This can include services that track/detect changes in the objects descriptions/metadata as well as in the learner model specifications. It may need to call other services that translate changes detected from one schema to another as well as other access services. In its general form it creates events to publicise anything of potential interest to other services, such as changes in the states of an object, the activation of an application, the modification of a Web page, changes in learner behaviour etc.</td>
</tr>
</tbody>
</table>
2.2 Data Model Specifications for Personalisation Services

E-learning resource metadata standards play an important role in this approach to specify learning materials (learning objects metadata, such as the IMS-LOM), courses (course metadata, such as the CDM XML schema), and user models (user metadata, such as the IMS-LIP, or eduPerson). They are used to semantically annotate educational material of all kinds and personalise the content, and provide interoperability, reusability and accessibility.

Let us take for example the case of user metadata. Depending on the application domain and the implementation approach, the provision of personalisation requires creating and updating a user model for each learner or for each learner group, where the dimensions of the different models may differ in their semantic descriptions. Although a variety of schemes is available for describing learner model attributes for various domains, such as OUNL-EML, PALO, PAPI, IMS, ARIADNE, there is no standard way to represent application-specific learner models (Godby, 2004; Heery and Patel, 2000).

Certain attributes of the learner model may require mixing, adapting and sometimes extending a data model to meet specific application requirements for the personalisation functionalities (Godby, 2004; Keenoy et al., 2004). In Figure 2, we provide a simplified view of a user model using an RDF-like representation. It seems that an RDF-enabled approach allows incorporating elements from several specifications in order to create an application-specific user model. Along this line, the schema of Figure 2 uses elements of the PAPI (Personal and Private Information) standard, the IMS (Instructional Management System) metadata specification, as well as application-specific elements. The Style element, shown in Figure 2, provides an example of how cognitive styles can be incorporated in the user model schema by extending the PAPI schema (or the catalog and entry elements of the IMS schema). The style taxonomy can include several cognitive style categorisations, such as Witkin; Honey-Mumford and so on; while the descriptor element can take values in the set of field dependent/field independent or reflector/theorist/activist/pragmatist respectively.

Moreover, the user schema may include elements of the eduPerson schema which defines how a subset of the user information might be represented in an enterprise directory. Whilst, the IMS specification defines application independent structured data models for representing various pieces of user information, the eduPerson elements of the user schema allow authorised users and services to access information regardless of where or how the original information is stored.

2.3 Personalisation Engine

This mechanism allows integrating two important functions of the system: adaptivity, which is system-controlled, and adaptability, which will allow users themselves to customise the application. The combined action of these two functions drives the personalisation decisions of the system.
The various personalisation services interact to provide the complete functionality of personalised system. The rational behind the interactions can be controlled by a personalisation engine that operates on the basis of event-trigger-rules (Lee et al., 2004). In this strategy, an event is the state of an object. If the state changes, then if there is a client that has subscribed for this event, is notified, if the criteria set by the client are met. Trigger is a specification that links events with rules or rule structures. If the corresponding events are posted, and all the essential triggers are present, a rule is executed. A rule is a high-level specification of executable code that is related to one or more events. A rule is composed by conditions, actions and alternative actions and is fired if the event is triggered for process.

We can distinguish two kinds of adaptation, namely push and pull, based on the direction of inference in the system. Figure 3 shows an example of the event-trigger-rule service for the case of a learner that selects to view a specific category of educational material. In this example we see that the adaptation depends on the learner model; the adaptation used here is pull adaptation. The system gets a list of all the events within a time window and checks which kind of presentation of the links is preferred by the learner among a set of alternatives.

3. CONCLUDING REMARKS

There are many ways to implement adaptive behaviour in e-learning systems. This paper presented an approach that is built around a set of services. One key challenge of this approach is defining what components are needed and how they should be connected so that they have minimum dependencies in order to be recombined for different educational purposes. Another challenge is identifying what services should be offered. The paper discussed a generic architecture for this purpose and briefly analysed its basic components.

ACKNOWLEDGEMENT

The work is partially supported by the Engineering and Physical Sciences Research Council (GR/R92554/02).

REFERENCES