Unifying Adaptive Learning Environments: authoring styles in the GRAPPLE project

Paul De Bra, David Smits, Evgeny Knutov, Eva Ploum, Kees van der Sluijs

Eindhoven University of Technology, Department of Computer Science
PO Box 513, NL 5600 MB Eindhoven, The Netherlands
debra@win.tue.nl, {d.smits, e.knutov, k.a.m.sluijs}@tue.nl, e.l.m.ploum@student.tue.nl

Abstract. The GRAPPLE project (EU FP7) aims at delivering to learners an adaptive technology-enhanced learning (TEL) environment that guides them through a life-long learning experience. To this end a general-purpose adaptive learning environment is integrated with a variety of learning management systems (LMSs), offering access to both learning material and to user model information from any LMS. This demonstration will show that within GRAPPLE (and its adaptive learning environment GALE), authors can develop adaptive learning material in different styles ranging from fully hand-crafted to completely automatically generated.

Keywords: adaptive learning environment, learning management system, presentation generation, authoring, integration

1 Introduction

The GRAPPLE project aims at delivering to learners a technology-enhanced learning (TEL) environment that guides them through a life-long learning experience, automatically adapting to personal preferences, prior knowledge, skills and competences, learning goals and the personal or social context in which the learning takes place. The same TEL environment can be used/accessed at home, school, work or on the move (using mobile/handheld devices). GRAPPLE includes authoring tools that enable educators to provide adaptive learning material to the learners. Authoring includes creating or importing content, assigning or extracting meaning from that content, designing learning activities and defining pedagogical properties of and adaptation strategies for the content and activities. To ensure the wide adoption of adaptation in TEL the GRAPPLE infrastructure works together with Open Source and commercial learning management systems (LMSs). Several suppliers of LMSs are part of the GRAPPLE project team. A distributed user modeling framework helps end-users to stay in control of their user profile while at the same time allowing them to use the profile to get personalized access to learning applications offered through different LMSs by different organizations.

The main technological developments in GRAPPLE are the construction of a generic adaptive learning environment (the GRAPPLE adaptive learning environment is called GALE) and an infrastructure for sharing user model information between

LMSs and GALE using the GRAPPLE User Modeling Framework (GUMF). In addition to this authoring tools are being developed for a learning application's *domain model* and *concept relationships*, together forming the Conceptual Adaptation Model (CAM). Authoring is kept independent of the adaptive environment GALE, by using a system-independent intermediate language: the GRAPPLE Adaptation Language (GAL).

We have studied a number of existing adaptive learning environments in order to extract their main adaptation functionality needed in the new generic adaptation engine. This study has been reported as part of a recent paper [7]. The study compared AHA! [4] KBS Hyperbook [5], APeLS [2] and Interbook [1] and also considered the AHAM [3] reference model. It reveals that there are many commonalities between all systems, regarding their *domain model* (DM) structure (consisting of concepts and services or tools), semantic *relationships* used to connect DM elements, the use of *overlay models* for user modeling, and two predominant ways to *reason* about how events (such as the learner reading a page or taking a test) result in user model updates and how the user model state (changes) result in adaptation of both the *content* and the recommended *navigation* in the learning application. Some systems (for instance Interbook!) perform mostly *forward* reasoning, using *event-condition-action* (ECA) rules, always computing and then storing results, whereas other system (for instance KBS Hyperbook) use Bayesian networks to compute results (but not store them) by reasoning *backwards* to check which events occurred that would lead to a user model state.

In this paper/demo we concentrate on the way *information content* is created and integrated into a learning application. Presentations (pages) in existing systems can be either authored completely manually (like in AHA!) or can be "generated" from components, for instance *pagelets* in APeLS.

In this paper/demo we show a toy example and three ways in which the information content for and the presentation of this (exact same) learning application can be created, all within the same GRAPPLE environment. We do not show how the conceptual structure is created using the GRAPPLE authoring tools for defining the DM and CAM and translating it to GAL, and we do not show how presentation and navigation are made adaptive based on that conceptual structure. (That is subject of other publications to come about GRAPPLE.)

2 The MilkyWay learning example

The example we use in this demonstration is not intended to be a part of a real learning application but is only a demonstration of the different approaches taken to create (adaptive) learning applications.

In the toy example we are mainly teaching the learner about stars, planets and moons. Information pages about all these concepts are fairly similar, and pages about different planets are even more similar to each other, and so are pages about different moons. All pages can be created by hand (possibly making extensive use of copy-paste) but they can in principle also be generated automatically. Below we show what pages of the example may look like:

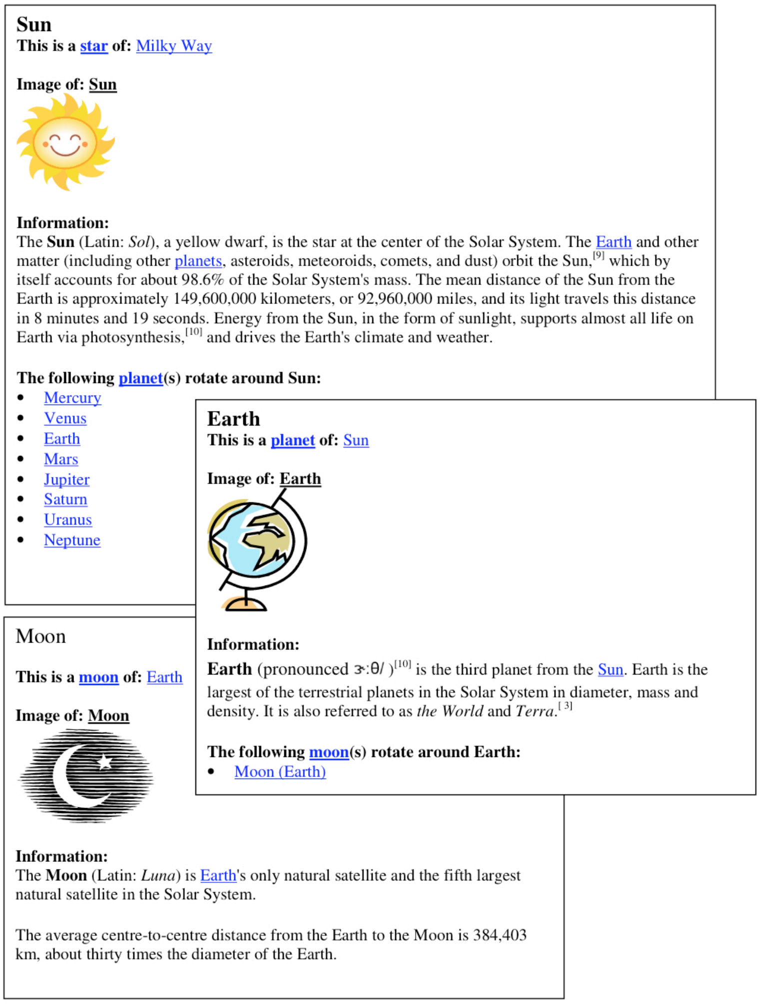

Fig. 1. Some pages from the MilkyWay example course (with content copied from Wikipedia).

As can be seen from the figure each page has more or less the same presentation of similar pieces of information:

- First there is a page title (which says what the page is about).
- In case of planets and moons the “parent” star or planet is named.
- There is an image of the star/planet/moon (which would be a real photographic image in a real course).
- The image is followed by a description of the star/planet/moon.
- In case of stars and planets their “children” are listed at the bottom of the page.

Because the image and description (and list of “children”) are different for every start/planet/moon it is tempting to write all pages by hand, following a common “template”. However, we will show two alternative approaches in the next section.

3 Three content and presentation authoring approaches

We consider the following authoring approaches:

- Hand crafted: in this approach an author writes all the pages of a course text. There need not be any common structure to the pages. Each page is associated with a concept in the domain model, and reading a page results in a corresponding user model update. The domain and user model may be used for content adaptation (conditional inclusion of fragments) and for link annotation based on prerequisites. To save space we do not show an example of the source of a hand-crafted page. Well-known examples of completely hand-crafted applications are the *hypermedia course* at the TU/e (currently at http://wwwis.win.tue.nl/2ID65/) and the AHA! tutorial (at http://aha.win.tue.nl/tutorial/).
- Template-based authoring: in this approach an author writes templates of the (few) different types of pages that occur in a course. But whereas in the hand crafted approach these template pages would be copied many times and filled in with the appropriate information, the template-based approach uses *references to* the information items rather than the items themselves. Each concept in the domain model has attributes either containing information or references to resources, like an image (of the star/planet/moon) or a text fragment (the description). The domain model can be structured as a hierarchy, which results in the relationships between a star and its planets and between each planet and its moons. Here is a snippet of the CAM for the example, written in the GRAPPLE Adaptation language GAL:

  ```
  Star_Unit rdf:type Unit [
            ...
            :hasAttribute [
               :name "starName";
               :label "This page is about star: ";
               :value [$Star/name]
            ];
            ...
  ```

  We see here that this “unit” has an attribute “starName” with a fixed text label “This page is about star: ” and a value which is taken from the domain model. In the template page for a star we can write something like:

  ```
  <h1>
            <variable expr="${starName/label}"/>
            <variable expr="${starName/value}"/>
  </h1>
  ```

  Through the authoring tools of GRAPPLE (not discussed in this paper) one can thus enter bits of content like the label and value for attributes that can be referred to in pages and automatically included. In the domain model the different stars are defined, each with their name, and this bit of GAL code and (X)HTML show how a template page for stars, associated with each star, will show the title and name for the correct star (looking like what is shown in Fig. 1). Likewise the image and description will be included, the parent object and the list of children if there are any. (The live demo will show the entire application based on just five templates: for milkyway, celestial object, star, planet and moon.)

- Generated: in the template-based authoring approach an author creates a single presentation template for each type of concept. Within the GRAPPLE project it is also possible to define adaptive presentation formats, in order to be able to adapt presentations for different output devices (mainly for smaller or larger screens). To this end it is possible to break up the template page into smaller units, each presenting one or more attributes of a concept (or the resource the attribute points to). In the MilkyWay example there are five potential such presentation units:
  1. The title (the label "This page is about: " and the name).
  2. The parent object's name.
  3. The image.
  4. The description.
  5. The list of children objects.

  For each of these units a small fragment of (X)HTML can be written, with code just like with the template-based approach shown earlier. In GALE the presentation can be controlled through adaptive layout hints associated with concepts. (The adaptive part of layout can be used to perform layout adaptation.) A layout defines a table structure (much like with HTML frames), associating a *view* with each table cell. A commonly use layout consists of 2 columns, one with a navigation menu and one with the information page.

```
<struct cols="200,*">
          <treeview />
          <mainview />
</struct>
```

  This layout would be used by the hand crafted and the template-based authoring approach. In the "generated" approach the <mainview> would be replaced for instance by a vertical <struct> containing 5 <fileview> elements, each including a fixed template fragment with content that is filled with information from the domain model. The elements can be *conditionally* included, so as to not show a parent for a star and not show children for moons.

  Note that this third approach towards authoring is typical in web-based information systems (WIS) research, for instance performed by the Hera research group [6]. A compiler from GAL to Hera does not yet exist but is certainly feasible. Whereas GALE can be used with all three content authoring approaches only the template-based authoring approach would work with Hera or other WIS platforms.

4 Conclusions

The GRAPPLE project not only tries to offer an adaptive learning experience to users of as many different learning environments (and LMSs) as possible, but also tries to accommodate different approaches to authoring the content and presentation of adaptive applications. This paper describes how applications can be created using completely hand-crafted pages, using template pages that are automatically reused for different information objects of the same type, and using completely generated pages and presentation formats.

During the demo we will show the complete MilkyWay example in all three authored forms, all running in the GALE adaptive learning environment, and all looking the same and behaving in exactly the same way. This will show that GRAPPLE brings seemingly incompatible approaches to authoring adaptive applications much closer to each other.

Acknowledgement

This paper and demonstration is part of the ongoing research and development in the EU FP7 project GRAPPLE (215434) and could not be realized without the close collaboration between all 15 GRAPPLE partners, not listed as authors, but nonetheless contributing to the ideas described here.

References

1. Brusilovsky, P., Eklund, J. and Schwarz, E.: Web-based education for all: A tool for developing adaptive courseware. In: Computer Networks, vol. 30 (1-7), pp. 291-300, 1998.
2. Conlan, O., Wade, V., Bruen, C., Gargan, M.: Multi-Model, Metadata Driven Approach to Adaptive hypermedia Services for Personalized eLearning. In: Proceedings of the Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems. Lecture Notes in Computing Science 2347, Springer-Verlag, pp. 100-111, 2002.
3. De Bra, P., Houben, G.-J., Wu, H.: AHAM: A Dexter-based Reference Model for Adaptive Hypermedia. In: ACM Hypertext'99: Proceedings of the 10th ACM Conference on Hypertext and Hypermedia, pp. 147-156, 1999.
4. De Bra, P., Smits, D., Stash, N.: The Design of AHA!, In: ACM Hypertext'06: Proceedings of the 17th ACM Conference on Hypertext and hypermedia. pp. 133-134, 2006.
5. Henze, N., Adaptive Hyperbooks: Adaptation for Project-Based Learning Resources. Thesis (PhD). University of Hannover, 2000.
6. Houben, G.J., van der Sluijs, K, Barna, P., Broekstra, J., Casteleyn, S., Fiala, Z., Frasincar, F.: Hera. Book Chapter in: Web Engineering: Modelling and Implementing Web Applications, G. Rossi, O. Pastor, D. Schwabe, L. Olsina (Eds), Chapter 10, pp. 263-301, Human-Computer Interaction Series, Springer, 2008.
7. Knutov, E., De Bra, P., Pechenizkiy, M.: AH 12 Years Later: a Comprehensive Survey of Adaptive Hypermedia Methods and Techniques. In: New Review of Hypermedia and Multimedia, vol. 15, nr. 1, pages pending, Taylor & Francis, 2009.