H AH 2008

5th International Conference on

Adaptive Hypermedia and Adaptive Web-Based Systems

Hannover, Germany 29 July - 1 August 2008

Authoring of Adaptive and Adaptable Hypermedia

Workshop Proceedings

Organizers

Alexandra Cristea, *University of Warwick, United Kingdom*
Rosa M. Carro, *Universidad Autonoma of Madrid, Spain*

A3H: Sixth International Workshop on Authoring of Adaptive and Adaptable Hypermedia

Alexandra Cristea[1] and Rosa Carro[2]

[1]The University of Warwick, Department of Computer Science
Gibbet Hill Road, CV4 7AL, Coventry, United Kingdom
acristea@dcs.warwick.ac.uk
[2]University Autonoma of Madrid, Escuela Politécnica Superior, Spain
Rosa.Carro@uam.es

Abstract. This workshop follows a successful series of workshops on the same topic: A3H at UM'07, in Corfu, AH'06, in Dublin, at AIED'05, in Amsterdam, at AH'04, in Eindhoven, and at WBE'04 in Innsbruck. The current workshop focuses on the issues of design, implementation and evaluation of general Adaptive and Adaptable (Educational) Hypermedia, with special emphasis on three major trends in web authoring and the design and creation of web material: the introduction and use of *standards*, especially *Semantic Web related standards for authoring*; *automatization* processes in authoring; use of *Web 2.0* concepts and methodology in authoring.

Keywords: LAG; AHA; Grammar; Educational Adaptive Hypermedia; Adaptation; Adaptation Engine.

1. Short workshop description

Authoring of Adaptive Hypermedia has been long considered as secondary to adaptive hypermedia delivery. However, authoring is not trivial at all. There exist some approaches to help authors to build adaptive-hypermedia-based systems, yet there is a strong need for high-level approaches, formalisms and tools that support and facilitate the description of reusable adaptive websites. Only recently have we noticed a shift in interest (fuelled in part by this workshop series), as it became clearer that the implementation-oriented approach would forever keep adaptive hypermedia away from the 'layman' author. The creator of adaptive hypermedia cannot be expected to know all facets of this process, but can be reasonably trusted to be an expert in one of them. It is therefore necessary to research and establish the components of an adaptive hypermedia system from an authoring perspective, catering for the different author personas that are required. This type of research has proven to lead to a modular view on the adaptive hypermedia. Therefore, important issues to discuss are, among others: *Can adaptive hypermedia be authored based on standards only, or to what extent is this possible? How and to what extent can Semantic Web standards be applied in the* *authoring process? How can the authoring steps be done automatically? Which steps can be done in automatic way, and which steps need to remain manually authored? How can Web 2.0 concepts and methodology be used in authoring? How can semantics and ontologies be extracted from folksonomies in a useful manner? How can we support adaptive (or, in the educational field, pedagogic) scenarios? How can applying Semantic Web ideas, technologies and techniques support the adaptation scenarios? How can adaptivity be applied if authoring is done via Web 2.0 methods? How can the adaptive knowledge be formulated in a reusable manner? Are there any recurring patterns that can be detected in the authoring process generally speaking? How does grouping of authors or assignment of authoring roles influence the authoring process, and are there ways to optimize this?*

Major Themes are thus: *Authoring for adaptivity based on Semantic Web standards or other standards; Authoring for adaptivity based on Web 2.0 ideas, methodology, technology, applications; Folksonomies and Ontologies in Authoring of Adaptive Hypermedia; Automatization in Authoring of Adaptive Hypermedia; Design patterns for adaptive hypermedia; Authoring group user models for adaptive/adaptable hypermedia; Authoring in groups; Role-based authoring; Automatic, adaptive authoring; Authoring pedagogic models for adaptive/adaptable educational hypermedia; Generic authoring for adaptive/adaptable hypermedia; Generic authoring tools in adaptive/adaptable hypermedia; Reusable user models, group user models, and pedagogic models; Evaluation of authoring tools for adaptive hypermedia; Evaluation of adaptive hypermedia design patterns; Evaluation of adaptive hypermedia authoring patterns.*

2. Conclusions

The workshop is expected to lead to a better understanding and cross-dissemination of how Semantic Web technology, standards and methods, as well as Web 2.0 methods can be applied in the field of Authoring of Adaptive and Adaptable Hypermedia and Web Personalization. Moreover, this workshop will lead to a better understanding of how manual and automatic authoring can be combined to achieve best results, especially based on patterns extracted from existing design and authoring processes in AH.

The workshop aims to attract the interest of the related research and practitioner communities to the important issues of design and authoring, with special focus on Semantic Web and Web 2.0 applied in adaptive hypermedia; to discuss the current state of the art in this field; and to identify new challenges in the field.
Moreover, the workshop can be seen as a platform that enables the cooperation and exchange of information between European and non-European projects.

Acknowledgments. This research has been performed with the help of the EU ALS Minerva project (Adaptive Learning Spaces) and the GRAPPLE EU FP7 project.

References

1. Baldiris, S., Santos, O. C., Moreno, G., Huerva, D., Fabrega, R., Boticario J.G., Management of learning styles, competences and access device preferences to alleviate the authoring of standard-based adaptive learning designs, 6th Int. Workshop on Authoring of Adaptive and Adaptable Hypermedia at the 5th Int. Conf. on Adaptive Hypermedia and Adaptive Web-Based Systems, 29 July 2008, Hannover, Germany.
2. Dhawan, S., Early Design Performance Prediction for Authoring Web-based, 6th Int. Workshop on Authoring of Adaptive and Adaptable Hypermedia at the 5th Int. Conf. on Adaptive Hypermedia and Adaptive Web-Based Systems, 29 July 2008, Hannover, Germany.
3. Ghali, F., Cristea, A.I., Stewart, C.D., Hendrix, M., Collaborative Adaptation Authoring and Social Annotation in MOT, 6th Int. Workshop on Authoring of Adaptive and Adaptable Hypermedia at the 5th Int. Conf. on Adaptive Hypermedia and Adaptive Web-Based Systems, 29 July 2008, Hannover, Germany.
4. Ghali, F., Sharp, M., Cristea, A.I., Folksonomies and Ontologies in Authoring of Adaptive Hypermedia, 6th Int. Workshop on Authoring of Adaptive and Adaptable Hypermedia at the 5th Int. Conf. on Adaptive Hypermedia and Adaptive Web-Based Systems, 29 July 2008, Hannover, Germany.
5. Gutierrez Santos, S., Losa, F.J., Kloos, C.D., Towards a formalization of the automatic generation of exercises, 6th Int. Workshop on Authoring of Adaptive and Adaptable Hypermedia at the 5th Int. Conf. on Adaptive Hypermedia and Adaptive Web-Based Systems, 29 July 2008, Hannover, Germany.
6. Hendrix, M., Cristea, A.I., A meta level to LAG for Adaptation Language re-use, 6th Int. Workshop on Authoring of Adaptive and Adaptable Hypermedia at the 5th Int. Conf. on Adaptive Hypermedia and Adaptive Web-Based Systems, 29 July 2008, Hannover, Germany.
7. Melia, M., Pahl, C., Towards the Validation of Adaptive Educational Hypermedia using CAVIAr, 6th Int. Workshop on Authoring of Adaptive and Adaptable Hypermedia at the 5th Int. Conf. on Adaptive Hypermedia and Adaptive Web-Based Systems, 29 July 2008, Hannover, Germany.
8. Ortigosa, A., An adaptive hierarchical questionnaire based on the Index of Learning Styles, 6th Int. Workshop on Authoring of Adaptive and Adaptable Hypermedia at the 5th Int. Conf. on Adaptive Hypermedia and Adaptive Web-Based Systems, 29 July 2008, Hannover, Germany.
9. Ras, E., Rech, J., Weber, S., Collaborative Authoring of Learning Elements for Adaptive Learning Spaces, 6th Int. Workshop on Authoring of Adaptive and Adaptable Hypermedia at the 5th Int. Conf. on Adaptive Hypermedia and Adaptive Web-Based Systems, 29 July 2008, Hannover, Germany.
10. Razmerita, L., Modeling Behavior of Users in Semantic-enhanced Information Systems: The role of a User Ontology, 6th Int. Workshop on Authoring of Adaptive and Adaptable Hypermedia at the 5th Int. Conf. on Adaptive Hypermedia and Adaptive Web-Based Systems, 29 July 2008, Hannover, Germany.
11. Saloun, P., Velart, Z., Adaptive ontology-based navigation, 6th Int. Workshop on Authoring of Adaptive and Adaptable Hypermedia at the 5th Int. Conf. on Adaptive Hypermedia and Adaptive Web-Based Systems, 29 July 2008, Hannover, Germany.

Table of Contents

Folksonomies and Ontologies in Authoring of Adaptive Hypermedia

Fawaz Ghali[1], Mike Sharp[2], and Alexandra I. Cristea[1]

[1] Department of Computer Science, University of Warwick,
Coventry, CV4 7AL, United Kingdom
{F.Ghali, A.I.Cristea}@warwick.ac.uk
[2] School of Engineering, University of Greenwich,
Chatham Maritime, Kent, ME4 4TB, United Kingdom
M.W.Sharp@gre.ac.uk

Abstract. *Semantic Web* and *Social Web* are two rapidly growing areas, evolving independently but complementing each other. On one hand, ontological aspects in the Semantic Web represent the top-down model, which lacks flexibility and scalability, in addition to facing the technical challenges of deployment on the current web. On the other hand, folksonomies on the Social Web represent the bottom-up model, which consists of unstructured data and carries no semantics, leading to questions of accuracy and reusability. Therefore, we worked on merging folksonomies from the Social Web with ontologies from the Semantic Web. This merge has the following advantages: 1) Creating semantic relations between tags of folksonomy; 2) Enabling reasoning on the Social Web. 3) Augmenting the authoring process of adaptive hypermedia, by providing rich, free, but also hierarchically structured data from the combined Social and Semantic Web.

Keywords: Semantic Web, Social Web, ontology, folksonomy, semantic enrichment, authoring, adaptive hypermedia.

1 Introduction

Adaptive and adaptable hypermedia authoring is challenging, especially with respect to the move to the Social Web and/or the Semantic Web. The *Semantic Web* is a web of data, in which the machines are able to understand, leading to an effective way of finding and sharing information [23]. Moreover, in [6] we find that "the Semantic Web is all about authoring", in the sense of rigorous, rich creation of annotated data. The *Social Web* is also based on annotations, however, free ones. The term refers to the activities of users on the web, such as chatting, discussion groups, and online communities; these activities are supported by social network services, which collaborate to make the web an open social network [22]. The Semantic Web organizes and categorizes data into *ontologies*, a more rigorous, formal way of creating machine-processable data structures. On the other hand, the Social Web uses folksonomies, which represent a method of collaborative categorization using freely-chosen keywords called *tags* [19]. In this paper, we present a mechanism of mapping unstructured data from Web 2.0 to structured ontologies. This further allows augmenting the authoring of adaptive hypermedia by providing rich, free, but also hierarchical structured data from the social and semantic web. This can be used by authors in different ways. Either to add adaptivity to linear social web data, or as a pool of keywords to draw from, as will be explained. Such automatic authoring steps can be a help in the "*difficult problem*" [7] of adaptation authoring.

2 Merging Methodology

Our social and semantic web merging methodology consists of three main phases: 1) *Filtering misspelled* tags from the Social Web. 2) *Grouping* unstructured tags based on co-occurrence values. 3) *Mapping* grouped tags onto matching elements of ontologies (using Swoogle [21] and Jena [11]). Fig. 1 illustrates these main phases. The input is a set of unstructured tags from Flickr. These are filtered using the Google API[1]. Next, they are grouped, using the relations between these tags (i.e., their mutual *co-occurrence values*, as explained in section 2.2). Thirdly, we used Semantic Web tools (i.e., Swoogle and the ontologies available on the web) for the mapping process between grouped tags and elements of selected ontologies. Finally, we create a structured hierarchy from the structured tags. Fig.1. also illustrates that our work is divided into two sections; the first section is done on the Social Web, whereas the other section is done on the Semantic Web.

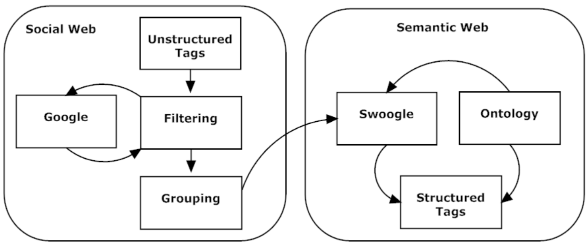

Fig. 1. Merging Social Web with Semantic Web.

2.1 Filtering Misspelled Tags

Tags on Flickr are generated in abundance by end users. However, the quality often suffers. E.g., some of those tags are misspelled. We used the Google API to correct misspelled keywords, where the Google spell checker software analyses the query, examining if the tag represents the most used version of a word. If it finds more related search results with an alternative spelling, based on occurrences of all keywords on the Internet, then the misspelled tag is replaced with the suggested one.

[1] API: Application Programming Interface; Google API: http://code.google.com/

2.2 Grouping Filtered Tags

In this phase, we group similar tags based on statistical information retrieved from their *co-occurrence values*, by using the Flickr API. The *co-occurrence value* between two tags represents how many times these two tags are used together in tagging multiple resources. A formalization of a classical formula for computing the co-occurrence is shown below:

$$\text{co-occurrence}(tag1, tag2) = \sum_{res \in resources} istag(tag1, res) * istag(tag2, res) \qquad \textbf{(1)}$$

$$istag(tag, res) = \begin{cases} 1, tag \in tags(res) \\ 0, rest \end{cases}$$

tags(res) – all tags of resource *res*; *resources* – the set of all resources;

For example, consider that the algorithm found two images on Flickr tagged with "love" and "harmony". Then, the co-occurrence value between the tag "love" and the tag "harmony" is 2. If these two tags are used together again, the co-occurrence value will be incremented accordingly. This means that the relatedness between the two tags is determined by the number of the times these two tags appeared together in the whole dataset.

2.3 Mapping between Grouped Tags and Elements of Ontologies

Grouping tags is a first important step. However, the inclusion of a number of tags in the same group does not provide any information about the type and structure of the relation between these tags. Therefore, in order to enrich the grouped tags with semantic relations, the next phase requires adding semantic content (i.e., from ontologies). Thus, matching between the grouped tags and ontologies is performed.

We used Swoogle [21] to retrieve the required ontologies. Swoogle is a semantic search engine for semantic contents on the web. It returns the ontologies as RDF files; however, it does not provide reasoning and extraction of fully automated semantic relations between the grouped tags. This can be performed with another semantic API, such as Jena [11]. Jena is an open source semantic web framework for Java. It provides an API to extract data from RDF and write to RDF graphs. The purpose of using Jena is to parse and serialize RDF files retrieved from Swoogle to determine the semantic relations between the tags within a group.

3 Experiment

We used data from Flickr, a social website, allowing users to upload their photos and describe them via "tags" (a form of metadata). The initial subset is called *tag cloud* of Flickr[2], which is the set of the most popular tags. This amounts to about **145 tags** (*size(tag_cloud)* = 145). However, this set of tags does not define relations between tags, or contain quantifiers of these relations, which are necessary to analyze the co-occurrence among the tags. We extract relations by extending the initial tag cloud to a

[2] http://www.flickr.com/photos/tags/

larger set of *related tags* (**11,138 tags**). Flickr has an open API[3] for data retrieval (e.g., photos, tags, profiles or groups).

3.1 Conditions of the Experiment

In the experiment, we combined a set of APIs from different sources to accomplish our goal of enriching tags on Flickr with semantics relations. In order to limit the results, we also set up a set of conditions as follows:

1) Each tag has to occur at least ten times to be a part of a group (this value is determined experimentally, and is based on the fact that spam tags and/or unrelated tags present in our database have a lower average of co-occurrence value, i.e., *co-occurrence (spam_tag, any_tag) < 10*).
2) Every two groups that share more than five tags must be combined into one group, to avoid redundancy (this result in an average of number of tags in per group of ten tags).

3.2 Steps of the Experiment

We developed a specific application that can work with the APIs mentioned in this paper. Firstly, we retrieved tags from the *tag cloud* and its related tags, and then we calculated the initial co-occurrences between those tags, using the appropriate Flickr API functions. Secondly, we filtered the misspelled tags using Google's spell checker API, as below. If the word is correctly spelled, the check spelling function will return NULL. Following, we calculate the final grouping. One group is created for each *tag cloud* tag. Matching the rest of the tags to these groups is based on a *co-occurrence value* threshold of 10 (based on an average experimental value). Next, the initial number of groups are reduced, by searching for common tags between groups. In the final phase of the experiment, we interpret relations between grouped tags by mapping them onto ontology elements.

3.3 Results

As previously mentioned, the initial input of the experiment was a set of **145 tags** from the *tag cloud*. Each of these tags is used to retrieve related tags. The extended resulting input was a set of **11,283 tags**. This set of tags was manipulated in the following three phases: 1) Filtering. 2) Grouping. 3) Mapping. During the filtering process, all misspelled keywords (e.g., "aple" instead of "apple") were corrected using Google API spell checker. **2,124 tags** were misspelled, **1,854 tags** of the misspelled tags were corrected, whilst the rest of **2,302 tags** were removed from the total set, because the co-occurrence value was below the threshold. Some tags appeared in different groups with different meanings (e.g., "food" appearing in both food and hobby group). This is solved in the mapping process, by mapping semantic

[3] http://www.flickr.com/services/api/

relations to different ontologies. After the filtering process, the remaining **8,710 tags** were categorized into groups, based on the *co-occurrence values* between tags.

To concretely show some results[4], we illustrate the processing of the tag "Food" from the tag cloud. For it, we retrieved a set of **71 related tags**. We filtered the previous set using the Google API spell checker, and after applying the conditions of our experiment, we obtained a set of **14 tags**, as follows: {**food**, **fruit**, **dessert**, **pasta**, cake, **red**, **seafood**, **fish**, **meat**, **grape**, **spaghetti**, **vegetable**, **bread**, **green**}. Finally, we mapped those 14 tags onto the elements (concepts, properties, or instances) of ontologies, using Swoogle (to retrieve ontologies that contain the selected tag, 'food') and Jena (to define relations among tags in this group by calling the appropriate function). Due the lack of semantic contents on the web, not all groups are mapped onto ontologies; however this group on this well-known concept, "food", mapped almost completely on the Food Ontology[5] (Please note that even the food group doesn't completely map onto the Food ontology. For instance, there is no mapping of the tag 'cake'). The list of the mapping results are too numerous to list in this paper, but here is a sample: (items in **Bold** are elements of the processed group 'Food', shown below as they map onto the Food ontology.)

```
{Food → Fruit}
{food:EdibleThing → Desert → food:SweetDessert}
{Desert → food:CheeseNutsDessert}
{food:EdibleThing → Pasta → food:PastaWithWhiteSauce}
{Pasta → food:PastaWithRedSauce}
{vin: property colour = Red}
{Meat → RedMeat → NonSpicyRedMeat}
{food:EdibleThing → seafood → food:Fish}
{RedMeat → NonSpicyRedMeat}
{seafood → food:Shellfish}
```

Hierarchical cluster analysis is based on a similarity matrix amongst tags. This matrix presents a table in which both rows and columns are the tags in same group and the cell entries are a measure of similarity (co-occurrence values) for any pair of tags. There are many measurement techniques for similarity between pairs of tags in the same group, such as (intervals, counts, binary), each with its own functions. For example, for the interval measurement, we can use Euclidean distance, Squared Euclidean distance, City block distance, Pearson correlation, and Cosine.

For within-cluster analysis and validation, the Count measurement using the Chi-square measure, the default for counting data, is most appropriate, as we compare the co-occurrence of the various tags within the cluster with the initial cloud tag that started the cluster. The Count Chi-square test in HCA compares observed counts of particular cases to expected counts (in our experiment, co-occurrence values between two tags). The result of the analysis can be depicted as *dendrograms*, also called hierarchical tree diagrams, which show the relative size of the proximity coefficients at which cases (tags) were combined (Figure 2). Each row represents a case (tag) on

---

[4] The result shown in this experiment is only a sample of the complete result, as there is no place to mention **8,710 tags**.

[5] http://www.w3.org/2001/sw/WebOnt/guide-src/food

the Y axis, while the X axis is a rescaled version of the proximity coefficients. Tags with higher similarity are close together, with a line linking them a short distance from the left of the dendogram. When, the linking line is to the right of the dendogram, the linkage occurs with a lower similarity coefficient, indicating that tags were agglomerated, even though much less alike (e.g. pasta and spaghetti are closely linked, but not as close as fruit and bread).

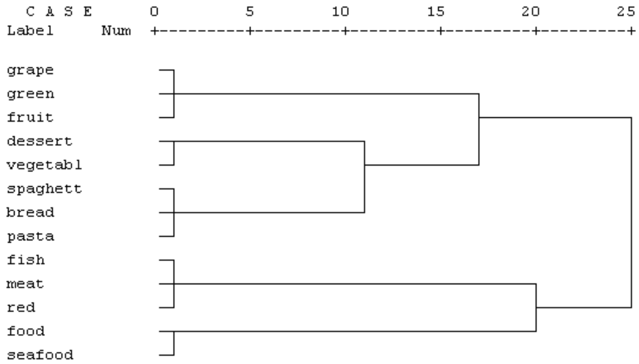

Fig. 2. Hierarchical cluster analysis for the food group

The previous dendogram presents a tree of visual classification of similarity (based on co-occurrence values), which illustrates another valid hierarchy. This hierarchy represents the bottom-up ontologies from folksonomies. In Figure 2, “green”, “grape”, and “fruit” are clustered first; similar to “fish”, “meat” and “red”; and “pasta”, “bread” and “spaghetti”; and so on. Next, these generated clusters (those are close to each other) are grouped together. The final result is a hierarchy which can be used to determine semantic relations among tags in same group, especially when the group does not have a corresponding ontology from the Semantic Web. It is thus possible, although not the main objective of this paper, to compare Web 2.0 bottom-up generated ontologies with expert-made, top-down ontologies from the semantic web. (e.g., the ‘Food’ ontology presented at the beginning of this section, with the resulting ontology in Figure 2). For here it suffices to note that our clustering method can thus be justified and analyzed via a statistical method independent to our initial clustering choice. However, our method is (arguably) computationally simpler to statistical methods, and thus preferable, as it is expected to scale better.

4 Discussions and Critical Analysis

We divided findings into three separate sections, reflecting our three processing phases, accordingly.

4.1 The Major Findings of Filtering Tags

The procedure outlined here has the effect of *decreasing the cost of creating and authoring semantic content*. We have obtained our set of tags 'for free', as the Social Web offers a great resource of data at a low cost.

Tags are context-specific, which means different users use the same tags with different meanings for different resources (e.g., user *A* uses the tag "mouse" for a photo of a type of rodent, whilst user *B* uses it to describe a computer input device).

Tags can be system-specific; we obtained our dataset from Flickr by using its API. However, other social websites exist, and it is reasonable to expect that the mapping results are different, as tag clouds can be different. For instance, extracting semantics from wiki content [1] requires using templates of Wikipedia only.

4.2 The Major Findings of Grouping Tags

All tags are connected to each other on the social web, thus, in principle, all tags belong to one set. However, this option is not practical for most application cases. However, the number of sets should be kept low. So, due to scalability issues, we initialized our set of groups with the elements in the tag cloud.

However, the number of sets should not be too low. Thus, as *large sets can lead to complex, time-consuming computations* and can contain too many false entries, we based the group inclusion condition on the *co-occurrence* value. This solution however could lead to deleting important tags from the group. Thus, there is a delicate balance between manageability and preciseness.

Not all social web tag-groups are covered by ontologies: some folksonomy tags have *no* matched ontologies. This finding is due to the lack of semantic content available on the web. A solution would be the building of ontologies from social data, e.g., via the users' behaviour mapped on the time dimension, as proposed previously.

Several groups share the same subset of tags, which leads to a high degree of similarity amongst those groups. The similarity leads us to merge those groups into larger groups, in order to avoid redundancy. Such similarity between groups may come, for example, from describing the same content in different ways. E.g., in the tag cloud there are two tags "Trip" and "Travel" whose groups have 7 shared tags: *trip* {travel, **vacation**, **sky**, **sea**, **water**, **beach**, holiday, nature, sun, ocean, **landscape**, sunset, road, clouds, light, street, boat, **Europe**, people, tree, family, red, night, friends, fun, California, island, car, parispark, roadtrip}; *travel* {**vacation**, nature, **sky**, **Europe**, **beach**, trip, **landscape**, **sea**, **water**, italy, architecture, blue, people, summer, mountains, paris, clouds, ocean, mountain, asia}.

4.3 The Major Findings of Mapping Tags

Since the experiment used tags from Flickr, we found that the most used ontology element was "instance" (not "concept" or "property"). The reason is that, when users tag a specific resource (here, pictures) they often tend to describe it with specific tags rather than more abstract concepts (e.g., names of people, models of cars, names of places, etc.).

Merging knowledge from multiple ontologies could provide a much richer perspective on the underlying semantics of tagging systems (as supported by result in [20]). Mapping to multiple ontologies can also help decrease the ambiguity level. For example, "mouse" (as a type of rodent or a computer input device) and can be mapped on two different elements of two different ontologies, clarifying the meaning. This also solves one of the main obstacles noted by [14], which focuses on polysemy (same word with multiple meaning) in a semantic collaborative tagging system.

It is possible to have tags in a group which are not mapped onto same ontology. This highlights differences between *usage* (folksonomy) and *semantics* (ontology).

The mapping process, compared to the related work [17], has the *advantage of automating* the *retrieving* ontologies from the semantic web, as well as *enriching* folksonomy tags with semantic relations.

5 Scenarios of Using the Authored Hierarchical Structure

Different scenarios can be applied to augment authoring environments with the produced hierarchy. The most straightforward is to import data from the social web into authoring environments, after it has been structured by mapping to ontologies, as shown in this paper. The imported data can then be used with different adaptation strategies, in order to personalize the display of the information. For instance, a simple rule would be 'show the current user all figures labelled with his current tag (from the user model (UM))':

```
IF (concept.tag == UM.tag) then concept.show = TRUE
```

Another scenario is using a structured keyword pool for *suggestions* or *correction*s, so that authors will start using more structured data in their keyword choice instead of free-chosen tags. This can be achieved as follows: 1) **Auto-replace**: feature found also in text editors or word processors. It involves automatic replacement of a particular string with another one, usually one that is longer and harder to type, such as "NYC" – unstructured tag from social web - with "New York City" – structured tag taken from the city ontology. 2) **Auto complete**: feature provided by many text editors and/or word processors where the system suggests a word or phrase that the user wants to type. This would require a similar processing of the keywords annotating concepts in authoring environments to the tags in the social web.

6 Related Work

Merging ontologies and folksonomies can benefit from the strengths of each of them [9]. The ontology provides the benefit of enabling semantic search queries, as well as applying reasoning on structured data. On the other hand, the folksonomy provides the benefit of flexibly generating tags. This merging utilises direct relations between tags based on *explicit* co-occurrence values rather than using complex mathematical relations between tags (as in [20]). Our work uses simpler co-occurrence computation methods that should allow for swifter processing, higher performance and scalability. Related work can be categorized in two main directions [18]: *1). Enrich the social web with semantic content*, as it is possible to apply the semantic enrichment between tags in folksonomy by using ontologies from the semantic web [17]. An example of semantic enrichment of tags are the weblogs in [3]; as well as the method of extending the (object, tag) pair in the folksonomy towards the triple (object, ontology, tag) in [12]. [15] aimed at extracting *structured* information from folksonomies, where the co-occurrence value among tags in folksonomies is the main factor to determine to which group the tag belongs to. 2). *Add information on users' activities or other social aspects to the semantic web*. Examples include the method of applying ontologies in a folksonomy model [2,4], or defining multiple labels for ontology nodes [13,8].

Our research is useful for both research directions, as it could both be used to enrich social web sites (such as Flickr) with semantic relations, or be used to correct and extend given ontologies with missing terms from the social web. Unlike the above presented related work, in our work we use explicit co-occurrence values among tags, which impacts on the performance, as the reasoning process is simplified. Moreover, we believe that co-occurrence values between tags are more important in the tag grouping process, rather then in the process of mapping onto elements in ontologies, especially when adding semantics to social web applications.

7 Conclusion and Future Work

The social web is driven by the power of its users, who collaborate and produce a massive amount of data. However, this data is unstructured and has no semantics, which leads to problems of retrieval accuracy and processing effectiveness. The semantic web is driven by the power of machine computing, where the computers are able to understand the data. Nevertheless, semantic concepts lack simplicity and scalability, and they are difficult to extend over the already existing large scale of information available on the current web. Our work thus aims at merging the Social Web with the Semantic Web by enriching tags of folksonomies with semantic relations. For this purpose, we used multiple APIs from different sources.

Concluding, the added value of our work is that it combines, in a simple, reproducible way, the strengths of the Social Web's bottom-up approach (i.e., the flexibility and simplicity of folksonomies) with the strengths of the Semantic Web's top-down approach (i.e., the accuracy and the hierarchy of ontologies). In addition, the work avoids the weaknesses in both Social and Semantic web, as the generated structures are extracted from folksonomies, rather than by applying predefined ontologies. For future work, we plan to investigate how to concretely integrate our work with authoring systems such as MOT (My Online Teacher) [5].

Acknowledgments. The work accomplished in this paper is supported by the Socrates Minerva ALS Project (Adaptive Learning Spaces) - 229714-CP-1-2006-1-NL-MPP and the GRAPPLE FP7 STREP (215434).

References

1. Auer, S. and Lehmann, J. Extracting Semantics from Wiki Content, The 4th European Semantic Web Conference 2007, Tyrol region of Innsbruck, Austria (2007)
2. Bateman, S., Brooks, C. and McCalla G. Collaborative tagging approaches for ontological metadata in adaptive e-learning systems. The 4th International Workshop on Applications of Semantic Web Technologies for E-Learning (2006)
3. Cayzer, S.: What next for semantic blogging? In Proceedings of the SEMANTICS 2006 conference, pp. 71–81, Vienna, Austria (2006)
4. Christiaens, S Metadata Mechanisms From Ontology to Folksonomy and Back, On the Move to Meaningful Internet Systems 2006 OTM 2006 Workshops, pp. 199-207, Montpellier, France (2006)
5. Cristea, A. and De Mooij, A. Adaptive Course Authoring: My Online Teacher, ICT'03, Papeete, French Polynesia, IEEE, IEE, ISBN: 0-7803-7662-5, pp. 1762-1769 (2003)
6. Cristea, A. I. What can the Semantic Web do for Adaptive Educational Hypermedia? Educational Technology & Society, 7 (4), 40-58. (2004).
7. Cristea, A., Smits, D., and De Bra, P.: Towards a generic adaptive hypermedia platform: a conversion case study. Journal of Digital Information (JoDI) Special Issue on Personalisation of Computing & Services, Vol 8, No 3. (2007)
8. Gruber, T. Folksonomy of Ontology A Mash-up of Apples and Oranges. First on-Line conference on Metadata and Semantics Research (MTSR'05) (2005)
9. Herzog, C., Luger, M. and Herzog, M. Combining Social and Semantic Metadata for Search in a Document Repository, 4th ESWC 2007, Innsbruck, Austria (2007)
10. J. B. MacQueen (1967): Some Methods for classification and Analysis of Multivariate Observations, Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, University of California Press, 1:281-297
11. Jena Semantic Web Framework, http://jena.sourceforge.net/ (accessed on 14/07/2007)
12. Lawrence, K. and Schraefel, M. Freedom and restraint: Tags, vocabularies and ontologies. 2nd IEEE Int. Conference on Information & Communication Technologies, Syria (2006)
13. Maedche, A. Emergent semantics for ontologies – support by an explicit lexical layer and ontology learning. IEEE Intelligent Systems - Trends & Controversies, pp. 78–86 (2002)
14. Marchetti, A., Tesconi, M., Ronzano, F., Rosella, M. and Minutoli, S. SemKey. A Semantic Collaborative Tagging System, 16th Int. WWW Conf., Banff, Canada (2007)
15. Mika, P. Ontologies are us: A unified model of social networks and semantics. The International Semantic Web Conference 2005, pages 522–536, Galway, Ireland (2005)
16. Monteroa, Y. and Solana, V. Improving Tag-Clouds as Visual Information Retrieval Interfaces, International Conference on Multidisciplinary Information Sciences and Technologies, InSciT2006. Mérida, Spain. October 25-28, 2006.
17. Sabou, M., Angeletou, S. and Specia, L. and Motta E. Bridging the Gap Between Folksonomies and the Semantic Web An Experience Report, 4th European Semantic Web Conference 2007, Tyrol region of Innsbruck, Austria (2007)
18. Schaffert, S. Semantic social software: Semantically enabled social software or socially enabled semantic web? Semantics 2006 conference, pp. 99–112, Vienna, Austria (2006)
19. Shelly, P. Television Disrupted: The Transition from Network to Networked TV. Focal Press. ISBN 0240808649. pp. 200 (2006)
20. Specia, L. and Motta, E. Integrating Folksonomies with the Semantic Web, 4th European Semantic Web Conference 2007, Tyrol region of Innsbruck, Austria (2007)
21. Swoogle Semantic Web Search Engine, http://swoogle.umbc.edu/ (accessed on 15/07/2007)
22. The Planetwork Journal, http://journal.planetwork.net/article.php?lab=reed0704 (accessed on 23/06/2007)
23. The World Wide Web Consortium, http://www.w3.org/2001/sw/ (accessed on 06/07/2007)

A meta level to LAG for Adaptation Language re-use

Maurice Hendrix and Alexandra Cristea

The University of Warwick, Department of Computer Science
Gibbet Hill Road, CV4 7AL, Coventry
United Kingdom
{maurice, acristea}@dcs.warwick.ac.uk

Abstract. Recently, a growing body of research targets authoring of content and adaptation strategies for adaptive systems. The driving force behind it is *semantics-based reuse*: the same adaptation strategy can be used for various domains, and vice versa. E.g., a Java course can be taught via a strategy differentiating between beginner and advanced users, or between visual versus verbal users. Whilst using an *Adaptation Language* (LAG) to express reusable adaptation strategies, we noticed, however, that: a) the created strategies have common patterns that, themselves, could be reused; b) templates based on these patterns could reduce the designers' work; c) there is a strong preference towards XML-based processing and interfacing. This has lead us to define a new meta-language for the LAG Adaptation Language, facilitating the extraction of common design patterns. This paper provides more insight into the LAG language, as well as describes this meta-language, and shows how introducing it can overcome some redundancy issues.

Keywords: LAG; AHA; Grammar; Educational Adaptive Hypermedia; Adaptation; Adaptation Engine.

1. Introduction

The use of adaptive systems [7] is increasingly popular. Commercial systems on the web (e.g., Amazon) or beyond (PDA device software) present at least a rudimentary type of adaptation. However, adaptation specification can not be fully expressed by standards[1] yet, and most commercial and non-commercial systems rely on proprietary, custom designed, system specific, non-portable, and non-interoperable adaptation. An intermediary solution, until standards emerge, is the creation of *Adaptation Languages*, which, with their power of semantics-based reuse, appear as a reliable future vehicle for all [8], [15]. Once written, the same adaptation strategy can be used for various domains. E.g., the strategy for *beginner-intermediate-advanced* written in the LAG language [8], could be used to teach students of varying knowledge level studying databases, mathematics or poetry. Similarly, the same domain model can be used with various adaptation strategies. E.g., a Java course can be taught via a strategy differentiating between beginner and advanced users, or between visual

[1] SCORM Simple sequencing allows basic adaptation. IMS-LD promises more for the future.

versus verbal users. However, there are a number of limitations regarding adaptation engines, which ultimately influence the efficient authoring of adaptation strategies, as based on an analysis of Interbook[2] [12] WHURLE[3] [14], AHA! [4], [5] and Personal Reader [1].

Thus, in this paper we define and analyze these limitations, illustrating them via a case study of a simple, yet powerful Adaptation Language, the *LAG language* [8]. Moreover, we propose a *meta-language*, as a supplement to LAG, showing how introducing it can overcome such limitations. Importantly, this solution is compatible with *extant* adaptation engines, instead of requiring the creation of new engines.

2. Adaptation Engine Issues and Limitations

The following are issues and limitations identified as influencing the authoring flexibility of adaptive hypermedia (AH) systems:

L1. Most adaptive hypermedia delivery systems determine the *adaptation on a per-concept base* [1]. A broad knowledge of the whole content at every adaptation step is (usually) unavailable, mainly due to run-time complexity limitations. Thus, adaptation strategies cannot specify complex inter-concept rules; e.g., a strategy with an arbitrary set of labels denoting topics of interest, displaying to the user concepts related to his topic, without limiting the possible topics at design-time.

L2. Adaptation engines don't (usually) allow for *non-instantiated program variables* [1]. Thus, authoring strategies which involve an unknown number of types, categories, etc., are currently not permitted. All domain-related variables need to be instantiated in the authoring stage.

L3. There are extreme difficulties arising when *combining multiple strategies* [1]. Adaptation engines usually update sets of variables based on some triggering rules, without knowing which high-level adaptation strategies these variables represent. An example of a combined strategy currently difficult to implement is one where the system checks whether the user prefers text or images, and then displays the preferred type of content, filtered via a beginner-intermediate-advanced strategy, where concepts are shown based on the user's knowledge.

In AHA! [4], [5] reasoning is mainly done on a per-concept base (for persistent attributes). Volatile attributes can contain expressions, which reference other attributes, allowing for backward reasoning. However, this does not fix problem L1 entirely. This method only allows for access to variables concerning concepts that have already been visited before or are in the same line of hierarchy. AHA! also does not allow for any free program variables (L2). AHA! can combine strategies (via the LAG language [8]) but does not offer any solution to conflicting naming (L3).

InterBook[4] [12] uses a knowledge-based approach to create adaptive, interactive electronic textbooks. Adaptation is more limited than in AHA!: it uses a classification of domain concepts into a *spectrum* and allows for adaptation towards the user's

[2] http://www.contrib.andrew.cmu.edu/~plb/InterBook.html
[3] http://whurle.sourceforge.net/
[4] http://www.contrib.andrew.cmu.edu/~plb/InterBook.html

current knowledge state. The prerequisites are computed on a per-concept base, and neither free variables nor combined strategies are at all possible (L1-L3).

In WHURLE[5] [14], the *lesson plan* specifies a path through the content *chunks*. Rules are defined on a per-concept base (L1), and no free program variables are allowed (L2) [13]. Multiple strategies are possible by using XML pipelines [16]. The issue of different strategies using conflicting naming (L3), however, remains.

Personal Reader [1] can deal with more sophisticated issues. It uses an RDF ontology with complex reasoning, so limitation L1 does not apply. However, it still does not offer free program variables (L2). Combining rules in an RDF ontology is less problematic, as multiple relationships can be defined at the same time. There are however limitations as to what can be implemented efficiently. For example, if we look at the OWL[6] ontology language (based on RDF), we see that although OWL Full is complete and has no limitations as to what can be expressed, only the very limited set of OWL Lite can be implemented efficiently. This however comes at the cost of a greater computational complexity, and therefore leads to a less scalable system.

3. A Case Study

3.1 The Theoretical Framework, in short: the LAOS Framework

In order to analyze the LAG language [8], a short briefing about the underlying theory is necessary. The LAG Adaptation Language instantiates the adaptation layer of the LAOS model [10]. The LAOS model is a general layered framework for Adaptive Hypermedia authoring, containing five layers: *Domain Model* (**DM**): with domains of content and their relations; *Goal & Constraints Model* (**GM**): filtering useful domain concepts (possibly from *multiple* domains) and grouping them; *User Model* (**UM**): with user specific variables, e.g. level, age, etc.; *Adaptation Model* (**AM**): defining how the content is adapted to users' needs; *Presentation Model* (**PM**): determining look & feel, navigation elements, as well as quality of service parameters.

3.2 Expressing Static Content in CAF (Common Adaptation Format)

The LAG Adaptation Language [8] processes information stored in CAF (Common Adaptation Format) [11]. CAF is an interfacing format that describes the static data needed for describing a Goal & Constraints Model (GM), and all the Domain Models (DM) it uses, ensuring that they all conform to LAOS [10]. Thus it defines concept maps, concepts, links and resources that are to be used in adaptation. CAF is mainly targeted at improving interoperability between different Adaptive Hypermedia systems, by offering a way to represent data in a system-independent way; e.g., CAF can be used to transport a GM and its related DMs between MOT [9] and AHA! [4]. CAF represents these models using a relatively simple XML format (see below).

---

[5] http://whurle.sourceforge.net/

[6] http://www.w3.org/TR/owl-ref/

```
<?xml version="1.0"?>
<!DOCTYPE CAF SYSTEM 'CAF.dtd'>
<CAF>
 <domainmodel>
  <concept>
   <name>Relational databases</name>
   <concept>
    <name>Fundamentals</name>
    <attribute>
     <name>theory</name>
     <contents>Relational theory</contents>
    </attribute>
   <concept>...
  </concept>...
 </domainmodel>
 <goalmodel>
  <lesson>
   <contents weight="0" label="beginner_title">
    Relational databases\Fundamentals\title</contents>
   <contents weight="0" label=" beginner_text">
    Relational databases\Fundamentals\theory</contents>
   <lesson>
    <contents weight="0" label="intermediate_title">
     Relational databases\Fundamentals\Definition\title
    </contents>...
   </lesson>...
  </lesson>
</CAF>
```

The example shown above represents a CAF file that contains one GM and one DM which this GM uses. The DM is called *Relational databases*. This Domain Model has one domain concept called *Fundamentals*. This concept has a domain attribute *theory* with the contents *Relational theory*. It also has other attributes, omitted due to lack of space. We see that the GM uses both the *title* and *theory* attributes of the *Fundamentals* concept of the *Relational databases* DM. It sets weights and labels for them, which, as we will see in section 3.3, are used by LAG [8] adaptation strategies. In short, titles and other elementary information (not shown here) are displayed to beginner students, and the theory is displayed to more advanced students.

3.3 A strategy in the LAG Adaptation Language

The Adaptation Language (LAG) [8] can express reusable adaptation strategies, describing adaptation, as prescribed by the Adaptation Model of LAOS [10]. As seen in section 3.2, items in the Goal & Constraints Model (GM) have weights and labels, which are used by the adaptation strategies. Below we show an example Adaptation Strategy described in the LAG language. This language works on structures defined by CAF, and thus is *domain specific*, with its domain being adaptive hypermedia in general; at the same time, within adaptive hypermedia, it is *generic*, as it can work with any content domain (e.g., databases, neural networks; chemistry, etc.).

The example below illustrates a simple strategy called '*beginner – intermediate - advanced*'. This strategy displays concepts to the user, depending on his experience level. The example uses the simpler labels 'beg', 'int' and 'adv' for concepts intended for beginner, intermediate and advanced users respectively (instead of the labels 'beginner_title', 'beginner_text', etc., as in section 3.2). The example also uses a number of variables. The 'show' variable, which determines if the concept is to be shown, is one of the few *core set variables* of the LAG language. Other variables are used, e.g., to record if a concept has been visited, or how many concepts of a particular group of concepts have been visited. It is more elegant to keep the set of variables as small as possible. Fewer variables make strategies smaller in terms of file size, thus easier to read, and in terms of memory usage, thus performing better.

The initialisation part (below) is performed only the first time the user enters the system; after that, every time the user selects a (lesson) concept, the implementation part (see comment 6), describing the actual interaction loop, is performed.

initialisation(

1) general: make every general (unlabeled) concept readable; mark every concept as "not visited yet" (beenthere =0);

```
while (true) (
   PM.GM.Concept.show = true
   UM.GM.Concept.beenthere = 0  )
```

2) initialize the number of concepts for beginning to advanced students to 0;

```
UM.GM.begnum=0 UM.GM.intnum=0 UM.GM.advnum= 0
```

3) count and store the actual number of concepts for beginner students;

```
  while GM.Concept.label == beg ( UM.GM.begnum += 1 )
  while (GM.Concept.label == beg) ( PM.GM.begnum +=1 )
```

4) count and store the actual number of concepts for intermediate students;

```
while (GM.Concept.label=int) (
   PM.GM.Concept.show = false
   UM.GM.intnum +=1 )
```

5) count and store the actual number of concepts for advanced students;

```
while (GM.Concept.label == adv) (
   PM.GM.Concept.show = false
   UM.GM.advnum += 1  )
```

6) set the level of the student to beginner, for the first entry in the system;

```
  UM.GM.knowlvl = beg  )
```

implementation(

7) UM.GM.Concept.beenthere computes the "number of times a Concept has been accessed". The following keeps track of how many <u>beg</u>inner, <u>int</u>ermediate and <u>adv</u>anced concepts still need to be visited. These rules are checked each time a concept is accessed. One concept is not 'aware' of other concepts, however.

```
if (UM.GM.Concept.Access==true) then (
  if (UM.GM.Concept.beenthere = 0) then
    if (GM.Concept.label == beg) then(
```

```
      UM.GM.begnum-=1      )
    if (GM.Concept.label == adv) then(
      UM.GM.advnum-=1      )
    if (GM.Concept.label ==int) then(
      UM.GM.intnum-=1      )
  UM.GM.Concept.beenthere+=1 )
```

8) Change the stereotype from beginner to intermediate; from intermediate to advanced when appropriate; make relevant concepts visible;

```
  if enough(UM.GM.begnum < 1
      UM.GM.knowlvl==beg,2) then(
        UM.GM.knowlvl = int        )    )
  if enough(UM.GM.begnum < 1
      UM.GM.knowlvl==int,2) then(
        UM.GM.knowlvl = adv        )    )
  if (GM.Concept.label == UM.GM.knowlvl) then(
      PM.GM.Concept.show = true    )    ) )
```

The strategy above illustrates a classical case of adaptation, to students of varying knowledge level[7]. The strategy works well because it 'knows' what labels to expect in the CAF file representing the Goal & Constraints model: 'beg', 'int', 'adv'. Currently, if other labels are also present, the conversion ignores them. However, what happens if we want to represent strategies with more complex labels, such as the ones in section 3.2 There, we had, e.g., various labels starting with 'beginner_' or 'intermediate_', but we didn't know in advance how many types of such labels exist. Still, we should expect to be able to perform some adaptive strategy and express it in the form of an adaptation program. As variables need to be instantiated, this introduces an intermediate step in the processing, as the next section shows.

4. Solutions to Adaptation Engines Issues and Limitations

Previously (section 3.3), we have seen an illustration of two of the current limitations listed in section 2: (L1) concept-based adaptation, where the same rule has to be copied in all concepts, and one concept doesn't (normally) affect other concepts directly, and (L2) the fact that adaptation engines don't allow for non-instantiated variables. A straightforward way of defeating these problems would be to build new adaptation engines. The first scenario could be achieved by establishing which labels exist, in the initialisation step. The second issue could be overcome by either allowing arrays of labels, or otherwise allowing multiple data to be stored in the label. However, in order to function with *current* systems, these issues should be solved in the authoring stage. For the third limitation (L3), the difficulty in application of multiple strategies, the MOT to AHA! converter, e.g., has already implemented an elegant solution (unique to our knowledge so far), in that it can apply multiple LAG files, with different adaptation strategies, with the order of execution set by priorities of the respective strategies (1: highest priority; any following number: lower priority):

[7] For examples of strategies please visit: http://prolearn.dcs.warwick.ac.uk/strategies.html

```
priority x   /* where x is a number */
```

Nevertheless, this method could override previous variables (e.g., if two strategies use UM.Concept.knowledge, only the update of the highest priority strategy counts). Thus, a *unitary strategy merge*, keeping track of all variables in use, based on multiple labels for domain-related concepts and attributes, is preferable. Moreover, many types of variables (e.g., arrays) are not allowed by Adaptation Languages, due to lack of adaptation engine support, limiting the adaptation that can be expressed.

However, we have noticed that a) strategies have common patterns, as has already been shown previously in [3], that could be reused; b) templates based on these patterns could reduce the designers' work; c) there is a strong preference towards XML-based processing and interfacing. Thus, XML-based templates should be used to move the processing to the authoring side and facilitate the extraction of re-use patterns.

For the creation of LAG files based upon a LAG template, explicit knowledge about the content is needed. CAF represents a flexible format for Adaptive Hypermedia content and is also used by AHA!. Therefore, a CAF file will be our choice for the content. For the LAG template files, the LAG files which follow the extended LAG description introduced in section 3.3 will be required. A pre-processor can replace the constructs added in section 5 by traditional LAG constructs. The resulting LAG file will then describe the same adaptation behaviour as the template LAG file, but for the specific labels encountered in the CAF lesson.

Implementing the pre-processor as a web-based application enables it to transfer both the unchanged CAF file as well as the resulting LAG file to the AHA! system, provided, of course, the appropriate rights are set and the pre-processor is on the same system as AHA! (currently AHA! only allows uploading files through a Java tool). To facilitate the use of multiple strategies, it should be possible to select multiple LAG templates. The user should be given a choice between creating the AHA! lesson and downloading the resulting LAG file. This process could, if the direct lesson creation is used, smoothly replace the current process, without requiring any extra effort from the user. This process is shown in the figure below.

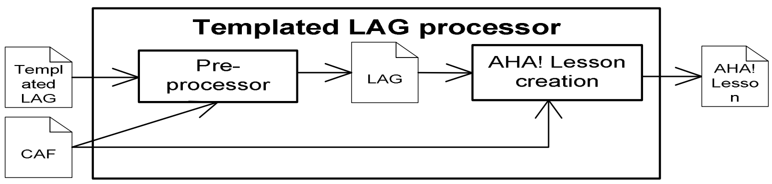

Fig. 1. System setup of template LAG Pre-processor.

5. Meta-level addition to LAG

To solve the limitations mentioned in section 2, we add, as said, a *pre-processing step* to the whole authoring process. This step takes a LAG template and the content, in the form of a CAF file, and pre-processes it. The result is a new LAG file which extends the strategy sketched by the LAG template for the specific content described in the CAF file. We want to accommodate future changes to LAG, as well as have our approach be reusable and easily implemented and maintained. Therefore we propose an XML-based notation for the template LAG files, while keeping the orginal LAG language unchanged for compatibility with current systems. Note that alternatively the changes could be incorporated into the Lag language directly but then it would lose its compatibility with existing systems. Since CAF is already written in an XML based notation, both documents can be used as input for an XSLT transformation which generates the resulting LAG file. Below we give the DTD (document type definition) for the template LAG file.

```
<!ELEMENT TLAG ((LAGfragment*, LIKE*)*)>
<!ELEMENT LIKE attribute CDATA value CDATA
(LAGfragment, MATCH, LAGfragment, (LAGfragment*, LABEL,
LAGfragment*)*) >
<!ELEMENT LAGfragment (#PCDATA)>
<!ELEMENT MATCH EMPTY>
<!ELEMENT LABEL EMPTY>
```

A template LAG file consists of a number of blocks of the following kind: a number of *LAG fragments* followed by a *LIKE* block. The LAG fragments contain LAG adaptation program snippets, similar to the examples showed in section 3.3. The LIKE blocks consist of an attribute and a regular expression against which it is matched, followed by a fragment of LAG program. The word *MATCH* represents the place where the *LABEL* needs to match the regular expression.

Below we show a fragment of the beginner-intermediate-advanced strategy. It shows how template LAG can be used to create an adaptation strategy that works with the CAF file example in section 3.2:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Server SYSTEM "tlag.dtd">
<TLAG> ...
  <LAGfragment>UM.GM.beginner_number= 0 </LAGfragment>
  <LIKE attribute='GM.Concept.label' value='*beginner*'>
      <LAGfragment> while(UM.GM.label= </LAGfragment>
   <MATCH/>
   <LAGfragment>)(UM.GM.beginner_number+=1)</LAGfragment>
  </LIKE> ...
</TLAG>
```

Following is an extract of the result of the pre-processing of a LAG template and the CAF file of the earlier example. The complete result is a LAG file, tailored towards the content of the CAF file. In the snippet below we see that the variable *UM.GM.beginner_number* is increased by one for each variable using the label *UM.GM.label.beginner_title* or *UM.GM.label.beginner_text.* These were exactly the labels matching the regular expression *'*beginner*'*. Applying (the DTD of) the LAG template solves some of the problems mentioned in section 2.

```
...
while
```

```
(UM.GM.label.beginner_title||UM.GM.label.beginner_text.
)(UM.GM.beginner_number+=1) ...
```

L1. *Problem: adaptation on a per-concept base*; a broad knowledge of the whole content at every step of the adaptation is (usually) unavailable.
Solution: such knowledge is not necessary in the adaptation engine. It is acceptable that this type of knowledge can be acquired as a one-off, at authoring time, as it is not to be expected that content labels will change at execution time. Therefore, the authoring strategy should contain this knowledge. As for an author it is difficult to manually extract all the pedagogical label types existent in a course, templates such as the DTD of the template LAG above can help in dealing with groups of labels (such as all labels containing 'beginner', i.e., '*beginner*'). An author can then generate the appropriate adaptation strategy (of which a snippet is shown above) in an easy and quick manner, making use of existing patterns in the authoring strategy itself.

L2. *Problem:* adaptation engines don't usually allow for *non-instantiated program variables*.
Solution: Unknown domain-related variables can be instantiated in the authoring stage, with the help of patterns specified via the LAG template language based on the above DTD. It is not necessary for an author to perform these searches manually; the two-step authoring system can extract unknown variables for him.

L3. *Problem:* the extreme difficulties arising when *combining multiple strategies*.
Solution: similar pattern extraction mechanisms have to be used in order to merge adaptation strategies. In (nearly) every system there is a limited number of weights and labels; this causes problems in combining a number of strategies greater than the number of weights and labels available. A solution to this can be to apply pattern matching on labels in order to be able to encode multiple strategies, by using the same label field. This thus enhances simple prioritization of strategies, as it allows the combination of multiple strategies which each requires specific labels.

6. Conclusions and further work

In this paper we have analyzed adaptation problems inherent in current adaptation engines, which reduce the power and generality of Adaptation Languages. We described and exemplified these issues with the help of the *LAG language*, currently one of the only exchange formats of Adaptation Language specification between systems. Moreover, we have moved one step further, by proposing improvements that can overcome run-time issues of adaptation engines, by solving them at the authoring stage. More specifically, templates can be used to create adaptation strategies, customized for the given domain models and pedagogical labels. For this purpose, we have proposed the *template LAG language*. The process is technically implemented by adding a pre-processor to the system setup, which has access to content at compile-time, which is not available at run-time. In such a way, more powerful adaptation strategies can be created for *existing* adaptation engines.

Acknowledgments. This research has been performed with the help of the EU ALS Minerva project (Adaptive Learning Spaces, 229714-CP-1-2006-NL-MINERVA-M) and the PROLEARN Network of Excellence.

References

1. AHA! Adaptive Hypermedia For All, http://aha.win.tue.nl
2. Abel, F., Brunkhorst, I., Henze, N., Krause, D., Mushtaq, K., Nasirifard, P., Tomaschewski, K.: Personal Reader Agent: Personalized Access to Configurable Web Services. ABIS 2006 - 14th Workshop on Adaptivity and User Modeling in Interactive Systems, Hildesheim, October 9-11 (2006)
3. Brown, E., Cristea, A., Stewart, C., and Brailsford, T. Patterns in Authoring of Adaptive Educational Hypermedia: A Taxonomy of Learning Styles, International Peer-Reviewed On-line Journal "Education Technology and Society", Special Issue on Authoring of Adaptive Educational Hypermedia, (Volume 8, Issue 3). (2005)
4. De Bra, P., Aerts, A., Berden, B., De Lange, B., Rousseau, B., Stanic, T., Smits, D., Stash, N.: AHA! The Adaptive Hypermedia Architecture. Proceedings of the ACM Hypertext Conference, Nottingham, Uk, August, 81--84 (2003)
5. De Bra, P., Smits, D., Stash, N., The Design of AHA!, Proceedings of the ACM Hypertext Conference, Odense, Denmark, August 23-25, 2006 pp. 133, and http://aha.win.tue.nl/ahadesign/, 2006.
6. Brailsford, T.J., Stewart, C.D., Zakaria, M.R. & Moore, A. (2002). Autonavigation, Links and Narrative in an Adaptive Web-Based Integrated Learning Environment.11th International World Wide Web Conference, Honolulu, Hawaii, 7-11 May 2002.
7. Brusilovsky, P.: Adaptive hypermedia, User Modelling and User Adapted Interaction, Ten Year Anniversary Issue, (Alfred Kobsa, ed.) 11 (1/2), 87--110 (2001)
8. Cristea, A.I., Calvi, L.: The three Layers of Adaptation Granularity. UM'03, Pittsburg, US. (2003)
9. Cristea, A.I., De Mooij, A.: Adaptive Course Authoring: My Online Teacher. Proceedings of ICT'03, Papeete, French Polynesia (2003)
10. Cristea, A.I., De Mooij, A.: LAOS: Layered WWW AHS Authoring Model and their corresponding Algebraic Operators. WWW03 (The Twelfth International World Wide Web Conference), Alternate Track on Education,Budapest,Hungary (2003)
11. Cristea, A.I., Smits, D., De Bra, P.: Writing MOT, Reading AHA! - converting between an authoring and a delivery system for adaptive educational hypermedia -, A3EH Workshop, AIED'05, Amsterdam, The Netherlands (2005)
12. Eklund, J., Brusilovsky, P.: InterBook: An Adaptive Tutoring System UniServe Science News Vol. 12. March 1999. 8--13 (1999)
13. Kostelník, R., Bieliková, M., Web-Based Environment using Adapted Sequences of Programming Exercises. In Proc. of Information Systems Implementation and Modelling - ISIM 2003. M. Beneš (Ed.). MARQ Ostrava, Brno, April 28-30, pp.33-40.
14. Moore, A., Stewart, C.D., Zakaria, M.R., Brailsford, T.J.: WHURLE - an adaptive remote learning framework, International conference on Engineering Education (ICEE-2003), July pp. 22-26, Valencia, Spain (2003)
15. Stash, N., Cristea, A.I., De Bra, P., Adaptation Languages as vehicles of explicit intelligence in Adaptive Hypermedia, In International Journal on Continuing Engineering Education and Life-Long Learning, vol. 17, nr 4/5, pp. 319-336, InderScience, 2007.
16. Zakaria, M.R., Moore, A., Stewart, C.D., Brailsford, T.J. (2003). "Pluggable" user models for adaptive hypermedia in education. Proceedings of the Fourteenth ACM Conference on Hypertext and Hypermedia, August 26-30, 2003, Nottingham, UK. pp 170-171

Towards a formalization of the automatic generation of exercises

Sergio Gutiérrez[2], Francisco J. Losa[1]
and Carlos Delgado Kloos[1]

[1] University Carlos III of Madrid
Avda. de la Universidad 30
E-28.911 Leganés, Madrid (SPAIN)
[2] London Knowledge Lab
Birkbeck College
23–29 Emerald St.
WC1N 3QS London (UK)

Abstract. An adaptable educative system needs a flexible assessment strategy in order to be able to adapt to many different kinds of students. However, creating a big amount of different exercises with different presentations, levels of difficulty, etc, is usually a very time consuming task. Many attempts have existed for creating big numbers of exercises automatically, but they tend to be strongly linked to some domain. This paper tries to extract some common foundation from them all to build a common framework, getting to a simple formalization of the concept of parametric exercise. The main idea is to have a core exercise that is different every time it is displayed to the student, being this change controlled by some parameters. The paper describes the four different types of parameters that exist in such an exercise, and presents an authoring tool and implementation that has been created according to these ideas.

1 Introduction

In every adaptive e-learning system, the process of assessment is extremely important. Assessment is needed to acknowledge the progress of the learners, both from the system and from the own learners' point of view: on one hand, checking their skills against a set of exercises allows the learners to become aware about their understanding of the concepts explained by the system; on the other hand, many processes on the system side depend up to some degree on the assessment process, like user modelling [1, 2] or sequencing [3].

There is a clear need for big sets of exercises in educational hypermedia systems. First, a limited or fixed set of tasks could be passed by the

learners simply by repeating them all a high number of times without any real learning [4]. Second, without a high number of different exercises or activities, the level of adaptivity that the system can offer is reduced. Unfortunately, creating a big set of tasks can be very time consuming [5]. Therefore, the advantages of making this process as automatic as possible, without any loss of expressive power, is evident.
It has been proposed that automatic generation of exercises should be a process directed by a set of parameters, leading to the concept of parametric exercises [6, 7] (henceforth, PE). The generation of the exercise thus becomes a function, whose input is a set of parameters and its output is the generated exercise. Several initiatives have been proposed in the last years (see Section 2), but they are usually bound to a particular domain. In the end, reuse of educational material is still a difficult task [8].
In this paper, the first step towards a formalization of the idea of PE is presented. We believe that a certain level of formalization is necessary to stablish a framework that would allow for the creation of generic tools to facilitate the automatic creation and correction of exercises in a reusable way. This paper draws lessons from previous work and tries to stablish some common foundation on which to build a general framework for automatic exercise generation, with a focus on modularity and reuse of the exercises. The contribution is modest, and the formalization is not complete. The actual implementation needs the exercise designer to have programming knowledge for all but the most basic exercises. This is a limitation for many teachers and designers without a programming background.
This paper is structured as follows. Section 2 shows some previous work on the field of automatic exercise generation. Section 3 describes the four types of parameters in a parametric exercise, while Section 4 shows the details of our current implementation using J2EE and AJAX. Section 5 closes the paper outlining open lines for future work and drawing the final conclusions.

2 Related Work

Work in automatic exercise generation is usually bound to a specific domain or task, making it difficult to generalise the results. For example, [6] points out that there are many systems for the teaching of maths or physics. In this kind of systems, the automatic generation of exercises is led by an arithmetic formula programmed in the background, all of whose parameters are given to the student according to a template; correction of the exercise consists on the trivial substitution on variables in the formula. However, the automatic generation of a high number of different exercises in other domains presents more challenging problems. Some initiatives for creating sets of different exercises automatically rely on the use of heuristics very limited to a specific domain and their ideas are difficult to generalize, like [9] for AC circuits, [10] for programming in Pascal or [11] for finite automata and Markov chains. Similarly, some ITS have modules that automatically create tasks according to some rules

based on the domain model [12]. The advantage of these systems is that they can create a high number of different exercises. Their main drawback is that they are usually very constrained to a particular domain, require a high degree of expertise on that domain, and the process of creating the rules is usually very time consuming [13].
The idea of using templates for creating diferent exercises has been used in maths and physics, but also in other domains. One example is [14], for programming. That solution, however, is only adequate for programming exercises, because the way in which the templates are defined is —in fact— a programming language, and probably not very intuitive for people without a technical background.
Tutoring systems for programming have shown good results generating multiple different exercises [6]. This is because they can rely on the use of language compilers to analyse the correctness of the solution provided by the student. This use of a general-purpose tool constitutes a great advantage. It means that complex exercises can be set in place with less effort than in other domains. However, this approach is inherently restricted to programming.
Fischer [15] proposes a way for creating exercises from an ontology of the model. Although the work is still in a preliminary stage, it can be considered as complementary to this paper. The ontology approach, with the usual graph-based visualization, might be a reasonable domain-independent compromise for the generation of PE. However, there are no parameters in Fischer's work at this point. This work focuses on the traversal of the ontology for the generation of the exercises and the correction of the answers.
Finally, it is worth noting that the IMS QTI specification provides some so-called *adaptive items*. However, they do not really address the problem of creating and correcting different exercises, they only allow an item to be scored adaptively over a sequence of attempts. The description of the exercise is fixed in QTI. Several systems have tried to overcome the limitations of the QTI specification like AthinaQTI [16] or MathQTI [17]. The latter uses a description language for the representation of interactive exercises which involve mathematical entities and have complex structures. Exercises are generated from static documents, in what can be seen as a layered scheme: the interactivity layer is applied over the static layer, yielding an interactive version of the content. This approach can also be seen as complementary of the one described in this paper.

3 Description of the parameters

As it has been already stated, PE can be instantiated several times; every time the actual exercise is different. In order for this to happen, there is a need to have several parameters that control how the exercise changes. Additionally, some parameters are also needed to allow the user to give answers to the exercise. Finally, if the exercise is to be integrated in some LMS that keeps track of the actions of the learner (e.g. for sequencing activities or for updating the learner's model), it has to give some information back to the system; this data can be viewed as well as a set of parameters. Therefore, a parametric exercise has four types of parameters: internal, external, answer and output. They are schematically represented in Figure 1.

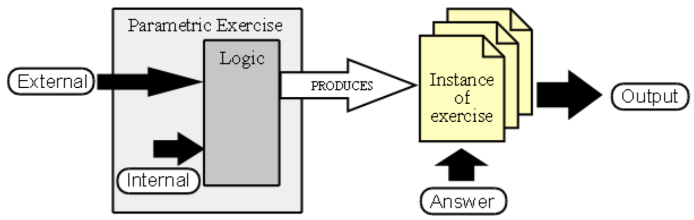

Fig. 1. Schematic description of the four types of parameters

External (or input) parameters (E) are parameters that have an influence in how the exercise is presented to the student (i.e. which instance is created). They can determine the data of the exercise, its presentation or any other characteristic. They are expected to be provided by the LMS before the exercise is actually instantiated. If they are not provided, they may have default values.
Internal parameters (I) have an effect on the instantiation of the exercise, but they cannot be specified by an external agent. Typically they will be defined randomly or according to some internal state of the machine running the exercise (e.g. internal clock).
Answer parameters (A) correspond to the answers of the student. They do not have an influence in the exercise, but are needed to correct and grade it. As a function of the correctness of the answers of the learner, the exercise may provide feedback after being graded. It is important to note that answer parameters are parameters of each particular instance of a problem, not of the general problem itself.
Ouput parameters (Ω) are data provided by the PE to the system in which the PE is running. They typically include information about the correctness of the answers of the student and the time needed to solve it. They do not influence the exercise or the feedback, but can have an influence on higher level structures (e.g. like deciding what is the next exercise to be shown).
PE usually include a description that is a function of the external and internal parameters. The effects on the exercise may vary: change the text provided, changing the actual data with which the learner must operate, providing different links to additional information (using techniques like link hidding [18]), etc. As an example, Figure 2 shows two different descriptions of an exercise involving algebra of polynomials. When an external parameter *find-roots* is set to `true`, the first version is shown; otherwise, the system displays the second one. This example illustrates how the description of an exercise can have an impact on its apparent difficulty depending on the learners' profile, without any need

of changing the actual data of the problem. Other parameters can control the challenge posed by the exercise by selecting a higher degree of the polynomial or limiting the time given for finding the solution. Another example, with a higher level of complexity, is presented in Section 4.

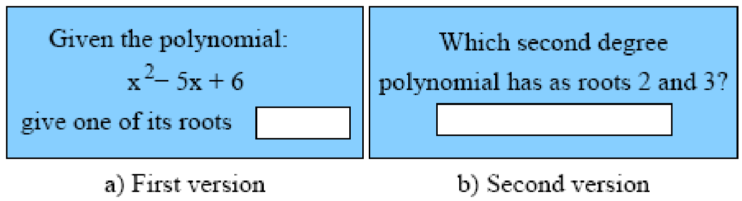

Fig. 2. Two different descriptions of the same exercise

PE may also provide some feedback on their topic. This is specially useful when the learner has failed on the exercise, but it may be valuable as well for students with a partial success. In our implementation, feedback works in PE in a similar fashion to QTI [19].

4 Authoring tool and sample example

We have implemented an AJAX-based authoring tool called ParEd [20] for testing these ideas. The tool eases the process of writing the description of the PE, setting the parameters, etc. It includes a client-side debugger that checks the PE for errors before it can be deployed in the LMS. Figure 3 shows an example of this mode: on the left, the editor is shown (parameters on the top, description and feedback in the middle, additional files at the bottom); on the right, the debugger complains about several mistakes (in this case, parameters are not correctly set, producing four errors). Errors detected by the debugger include: incompatibilities between the parameters, type errors in the parameters (they can be restricted to be integers, strings or booleans) and incorrect XHTML code.
In our implementation, the logic of the PE is specified in Java code. Every PE contains a Google Web Toolkit (henceforth GWT [21]) module that manages the adaptive aspects of its description (e.g. presentation). It also handles the exercise data generation and correction, communicating with the application logic stored on the server side through GWT RPC services. The application logic behind the data generation and correction is actually a separate Java class associated with the exercise, and is instantiated by the servlet on the server side using dynamic invocation.
There are cases in which the exercise designer may want the description of an exercise to contain more than just text and content (even adapted text through scaffolding [22, 23], etc). Therefore, in the GWT module of

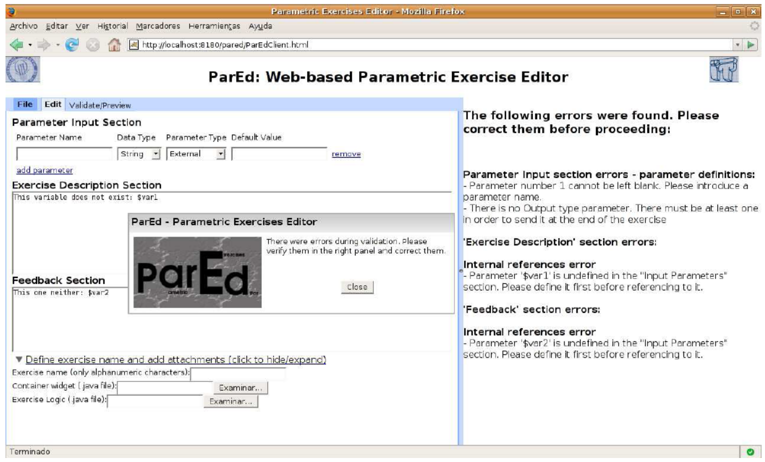

Fig. 3. ParEd debugging mode

our implementation of PE there is a widget called the *container*[3]. This container is inserted into the module and is able to represent complex exercise structures like tables of dynamic (i.e. parameter dependent) size and content, adaptive menus, interactive elements to click or drag and drop, etc. The container is not only able to present information to the student, but also gathers answers from answer fields, checkboxes, etc, and sends them to the server for correction. It is the designer's responsibility to create the container according to the logic that generates and corrects the exercise.

The logic contains the algorithms to generate the data of the exercise instances from the values of the parameters. It also includes the exercise correction engine. The logic runs in the server, so the produced exercises remain install-free and can be deployed in virtually any LMS.

ParEd includes a testing mode in which an instantiation of the exercise is created; then, after the "correct" button is pressed, it is corrected. Figure 4 shows this: the exercise instantiation is on the right, while all external parameters are on the left. Multiple instances of the PE can be created by selecting different external parameters and pressing the "Test values" button under the external parameters textboxes. Testing mode allows the exercise designer to debug the logic of the exercise, comparing inputs and outputs. Additionally, should a runtime error happen, it is automatically captured by the ParEd servlet and showed to the designer with a copy of the stack trace.

Several exercises have been created with ParEd, for two different domains (Computer Architecture and Operating Systems). In all cases, ParEd produces an AJAX implementation of a new exercise; this exercise is able to correct itself given the answers of the student. The use

[3] In terms of the GWT, the container is a descendant of *Composite*

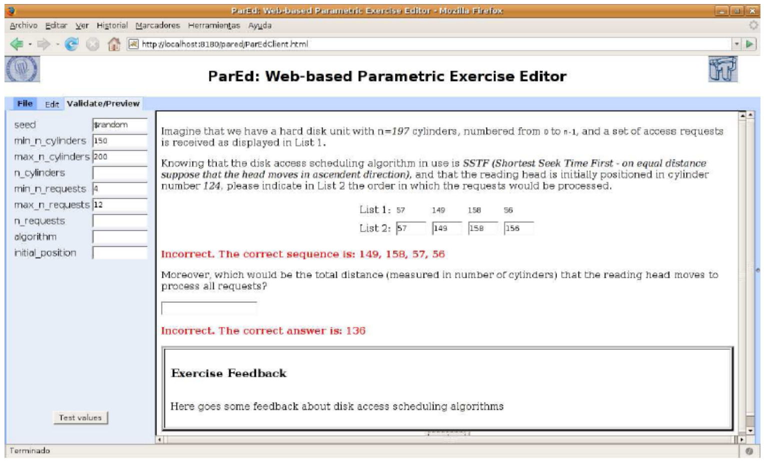

Fig. 4. ParEd testing mode

of AJAX provides a higuer level of *perceived* interactivity with the exercises on the part of the users. Other implementations of web exercises have used JSP pages [24], but this produces higher latencies. The AJAX paradigm reduces the communication between the client and the server to the specific parts of the page that needs to be adapted, and provides client-side functionality using Javascript. The combination of these two characteristics keeps the latency to a minimum.

One of the families of exercises created is presented here as an example of the flexibility of the approach. The exercise illustrates different disk access scheduling algorithms (e.g. FCFS, SSTF, SCAN, LOOK, etc). It reaches a complexity level well suited for engineering students attending a computer architecture course.

In the exercise description the students are presented with a set of pending disk access requests which are to be processed for a disk of a specific size, with its head placed at a certain initial position. The students are then requested to sort the requests following a specific disk access scheduling algorithm; additionally, they are asked to calculate the total distance travelled by the head of the disk after all the requests have been fulfilled.

This is an example of how a huge number of variations of the same exercise can be created to illustrate a set of concepts. The values of this exercise that are subject to change on each instance of the exercise (i.e. external parameters) are: the disk size (in number of cylinders), the number of requests, the value of the requests, the initial position of the disk head and the disk access scheduling algorithm. Additionally, it is possible to specify the minimum and maximum number of cylinders in the disk, the exact disk size, the initial position of the head unit, the minimum and maximum number of requests, a specific number of requests, and the disk access scheduling algorithm. The difficulty of this problem can be set by modifying the values of parameters such as the number of requests or the scheduling algorithm, as the complexity of the solution grows linearly with the number of requests. The virtually infinite number of different instances that can be produced makes it impossible to game the system [25, 26].

5 Conclusions and future work

A parametric exercise is an exercise that can be replicated an unlimited number of times with different data. This data makes the appearance, difficulty and solutions different on each instantiation, adapting them to the particular needs of every student. Parametric exercises allow for the creation of a high number of instances without any repetition, being very useful for training applications or for any web-based system that requires a lot of input about the knowledge of the student.

This paper has presented a first step towards the formalization of this concept, describing the four types of parameters that are relevant: external, internal, answer and output. The paper does not address other important processes like sequencing or user modelling; it is focused on the generation of the exercises. The definition of the four types of parameters is a small step in the process of creating a framework on automatic creation of exercises. Such a framework could open the way to a standardization effort. Standards are important because they promote reuse of learning material. In the case of static exercises, the IMS QTI specification has achieved a great level of success.

This paper has presented an implementation and authoring tool, programmed with J2EE and AJAX, that are used to illustrate the power of the proposal. The authoring tool allows to edit the exercises with a user-friendly web-based interface. It includes a debugging mode, that allows the designer to try the implementation of the exercise before it is deployed in the LMS.

Next steps in this research move along two main lines. The first one aims at making it easy for designers without a technical background to create complex exercises. Our current implementation requires programming skills on the part of the exercise designer. This restriction, common to most related systems, allows the exercises to go beyond simple formulae and creating complex challenges at the conceptual level like the one presented in Section 4. However, this is very limiting for us, as our goal is to create a system that could be used by people with little or no programming skills and for different domains (e.g. history or music). A possible solution may involve the use of script or iconic programming [27] languages, but this demands further investigation.

The second line of research studies how the PE paradigm can be used to improve specifications like QTI [19]. QTI is a succesful specification, but its support for adaptation is extremely poor. A complete formalization of the PE paradigm, probably overcoming the dependence on programming, may provide a contribution towards an adaptive questions and test specification.

Acknowledgments We would like to thank Dr. Manolis Mavrikis (London Knowledge Lab, UK) for his insightful comments on this paper.

References

1. Kobsa, A.: User modeling and user-adapted interaction. User Modeling and User Adapted Interaction **14** (2004) 469–475
2. Wahlster, W., Kobsa, A.: User models in dialog systems. In Kobsa, A., Wahlster, W., eds.: User Models in Dialog Systems, Springer (1990) 4–34
3. Gutiérrez, S., Pardo, A.: Sequencing in web-based education: Approaches, standards and future trends. In Jain, L.C., Tedman, R.A., Tedman, D.K., eds.: Evolution of Teaching and Learning Paradigms in Intelligent Environment, Springer (2007)
4. Valigiani, G., Biojout, R., Jamont, Y., Lutton, E., Republique, C.B., Collet, P.: Experimenting with a real-size man-hill to optimize pedagogical paths. In: SAC '05: Proceedings of the 2005 ACM Symposium on Applied computin g, New York, NY, USA, ACM Press (2005) 4–8
5. Murray, T.: Authoring knowledge-based tutors: Tools for content, instructional strategy, student model and interface design. Journal of the Learning Sciences **1** (1998) 5–64
6. Brusilovsky, P., Sosnovsky, S.: Individualized exercises for self-assessment of programming knowledge: An evaluation of quizpack. Journal of Educational Resources in Computing **5**(3) (2005)
7. Pardo, A.: A platform for parametrized exercises in web-based education. In: Int. Conf. of the Society for Information Technology and Teacher Education. (2002)
8. Ullrich, C., Chen, Q., Chen, S., Shen, L., Shen, R.: Investigating the reuse of course generation knowledge. In: Proc. of the 15th Int. Conf. on Computers in Education (ICCE2007), Hiroshima, Japan, IOS Press (2007)
9. Cristea, P., Tuduce, R.: Automatic generation of exercises for self-testing in adaptive e-learning systems: Exercises on ac circuits. In: 3rd Workshop on Adaptive and Adaptable Educational Hypermedia at the AIED'05 conference. (2005)
10. Kumar, A.: Dynamically generating problems on static scope. ACM SIGCSE Bulletin **32** (2000) 9–12
11. Tscherter, V.: Exorciser: Automatic Generation and Interactive Grading of Structured Exercises in the Theory of Computation (PhD Th.). Swiss Federal Institute of Technology (2004)
12. Anderson, J., Pelletier, R.: A development system for model-tracing tutors. In: Proc. of the International Conference on the Learning Sciences. (1991) 1–8
13. Murray, T.: Authoring intelligent tutoring systems: An analysis of the state of the art. International Journal of Artificial Intelligence in Education **10** (1999) 98–129
14. Azalov, P.: Synthetic exercises on the web. In: Int. Conf. on Information Technology: Coding and Computing (ITCC'05). (2005) 459–464

15. Fischer, S., Steinmetz, R.: Automatic creation of exercises in adaptive hypermedia learning systems. In: HYPERTEXT '00: Proceedings of the eleventh ACM on Hypertext and hypermedia, ACM (2000) 49–55
16. Lazarinis, F., Retalis, S.: Engineering an interoperable adaptive hypermedia testing tool supporting user adaptable strategies. In: IFIP Artificial Intelligence Applications and Innovations. (2006) 393–400
17. Mavrikis, M., Palomo, A.G.: Mathematical, interactive exercise generation from static documents. Electronic Notes in Theoretical Computer Science **93** (2004) 183–201
18. Brusilovsky, P.: Adaptive and intelligent technologies for web-based eduction. KI **13**(4) (1999) 19–25
19. IMS Global Consortium: IMS question and test interoperability specification (QTI). In: http://www.imsglobal.org/question/index.html (last accessed January 2008)
20. Llorente, F.J.L.: ParEd: Web-based Parametric Exercise Editor (M.Th). University Carlos III of Madrid (2007)
21. Chaganti, P.: Google Web Toolkit: GWT Java Ajax Programming. Packt Publishing (2007)
22. Jackson, S.L., Krajcik, J., Soloway, E.: The design of guided learner-adaptable scaffolding in interactive learning environments. In: Proceedings of the SIGCHI conference on Human factors in computing systems. (1998) 187–194
23. Collins, A.: Design issues for learning environments. In Vosniadou, S., DeCorte, E., Glaser, R., Mandl, H., eds.: International perspectives on the psychological foundations of technology-based learning environments, Lawrence Erlbaum Assoc. (1996)
24. Linillos, P.P., Gutiérrez, S., Pardo, A., Delgado Kloos, C.: Sequencing parametric exercises for an operating system course. In: IFIP Artificial Intelligence Applications and Innovation. (2006) 450–458
25. Baker, R.S., Corbett, A.T., Koedinger, K.R.: Detecting student misuse of intelligent tutoring systems. In: Proc. Int. Conf. on Intelligent Tutoring Systems. (2004) 531–540
26. Baker, R.S., Corbett, A.T., Koedinger, K.R., Roll, I.: Generalizing detection of gaming the system across a tutoring curriculum. In: Proc. Int Conf. on Intelligent Tutoring Systems. (2006) 402–411
27. Ichikawa, T., Hirakawa, M.: Iconic programming: Where to go? IEEE Softw. **7**(6) (November 1990) 63–68

Towards the Validation of Adaptive Educational Hypermedia using CAVIAr

Mark Melia and Claus Pahl

School of Computing, Dublin City University, Dublin 9, Ireland
mmelia@computing.dcu.ie

Abstract. Migrating from static courseware to Adaptive Educational Hypermedia presents significant risk to the course creator. In this paper we alleviate some of this risk by outlining how the CAVIAr courseware validation framework can be used to validate some pedagogical aspects in Adaptive Educational Hypermedia. To allow for this we present a novel method for interoperability in Adaptive Educational Hypermedia using Model Driven Engineering methodologies.

1 Introduction

The authoring of Adaptive Educational Hypermedia (AEH) is a major task for any course creator to undertake. The cost in time and effort leave many considering if the actual end product is cost-effective. Although recent advances in this area have been made, with the emergence of dedicated AEH authoring tools such as MOT [1] and the ACCT [2], there is still no way to check developed AEH for specific pedagogical problems.

Courseware validation is a design activity that automatically ensures the presence of certain structural and pedagogical characteristics in constructed courseware. Courseware validation allows the course creator to minimise the pedagogical problems which the learners must deal with when using immature courseware.

Using courseware validation in AEH, allows the course creator to automatically test the AEH for specific pedagogical problems, which may not be possible to check otherwise due to AEH's adaptive nature. This reduces the risk for the course creator, who wishes to migrate away from a static courseware and use AEH to deliver a course.

In this paper, we investigate how one AEH specification, the LAOS model can be validated using the Courseware Authoring Validation Information Architecture (CAVIAr) [3]. The paper firstly outlines the respective technologies, LAOS first and then the CAVIAr. Section 4 and 5 then introduces modeling technologies and methodologies and demonstrate how they are used to convert LAOS to CAVIAr for validation. Section 6 steps through the validation process, we conclude the paper in section 7 outlining our contribution.

2 MOT, LAOS and AEH Interoperability

The "My Online Teacher" (MOT) system [1], allows course creators to create adaptive courses using the LAOS conceptual architecture for adaptive hypermedia [4]. LAOS consists of five layered maps, where the higher layers are defined in terms of the lower layers. The layers are as follows starting with the lowest layer:

- **domain map** - "organises and structures the actual resources of the learning environment, as well as their intrinsic characteristics" [1].
- **goal and constraints map** - "this model filters, regroups and restructures the domain model, with respect to an instructional goal used to express educational goals" [1]. This is done by specifying the instructional weights of domain map concepts and by ordering the domain concepts.
- **user map** - used to specify the user knowledge, interests and learning styles.
- **adaptation map** - defines adaptive rules in terms of the lower layers. This map is defined using LAG, a 3-tier adaptive rule specification [5] .
- **presentation map** - defines course delivery environments variables, allowing the AEH to adapt to the delivery environment being used by the learner.

MOT is purely an AEH authoring environment, it does not allow for the delivery of AEH. In order for delivery of AEH material created using MOT must be delivered using an AEH delivery environments, such as AHA! [6] or WHURLE [7].

In order for the AEH developed using MOT to be delivered in an AEH delivery platform it must be interoperable with that delivery platform. To do this, Cristea et. al. makes the distinction between static and dynamic elements of the LAOS [1]. Static elements are exported from MOT through a common language, or lingua franca, known as the Common Adaptation Framework (CAF), which captures the domain map and the goal and constraint map. Dynamic elements, which describe the adaptive nature of the AEH and are captured using LAG.

MOT exports to CAF by converting the domain map and the goal and constraint map, which is stored are the MOT database, to the CAF XML specification, this can then be imported by the AEH delivery environment.

LAG captures the adaptation rules for AEH. The top level of the 3-tier LAG model is adaptation strategies, which are built on adaptation languages, which, in turn are built on direct adaption rules. In the LAOS context LAG direct adaptation rules are defined in terms of the lower layer maps. The LAG direct adaptation rules are IF-THEN or condition-action style rules, defined in a context-free BNF (Backus-Naur Form) style meta-syntax notation [1].

3 CAVIAr Courseware Validation

The CAVIAr is used in courseware authoring for automatic validation of a variety of courseware structural and pedagogical concerns including:

[1] http://wwwis.win.tue.nl/ acristea/MOT/help/LAGgrammar.doc

– inter-conceptual courseware sequencing - pedagogical concerns regarding the sequencing of concepts in courseware [8]
– intra-conceptual courseware sequencing - pedagogical concerns teaching one concept [9]
– the appropriateness of the type of learning material used at particular points in courseware
– courseware consistency
– elements of the instructional design in use in the courseware

The CAVIAr model allows for the course creator to identify instructional problems in the courseware prior to delivering it to learners. This is important as it allows the course creator to be confident that particular types of courseware problems are not present in the courseware developed. This allows formative evaluation of courseware to evaluate more complex pedagogical issues in the courseware.

Courseware validation using CAVIAr is achieved by modeling the courseware construction concerns. The CAVIAr uses a modeling structures very similar to that of LAOS, using four modeling layers as follows:

– **Domain Model** - a pedagogically neutral conceptual graph, used to structure knowledge to be covered in courseware
– **Learning Context Model** - Defines conceptual sequencing constraints and the learner stereotypes, each learner stereotype is defined as having assumed initial knowledge and a course goal in terms of the domain model
– **Courseware Model** - The courseware model is composed of two parts:
  • courseware structure, structured using courseware topics, where topics contains learning resources.
  • learning resource model, which contains a model representation of Learning Objects (LOs) and their metadata
– **Validation Model** - A constraint model which defines valid courseware

It is important to identify how the CAVIAr faclitates the representation of adaptive courseware, allowing for the mapping from AEH to CAVIAr. As we have outlined the courseware model defines the courseware structure and the LOs in the courseware. The courseware model is defined using a metamodel, an excerpt of which is in figure 1.

Adaptivity is achieved in a courseware model in two ways - specifying a "SEQUENCED_AFTER" relationship between two topics and by specifying an entryLearner requirement for a topic. The "SEQUENCED_AFTER" relationship allows the course creator to specify explicit sequencing constraints between topics. The entryLearner requirement allows the course creator to place a gate condition on a topic, so that the topic is only delivered to learners which satisfy the entryLearner requirement at any given point in time.

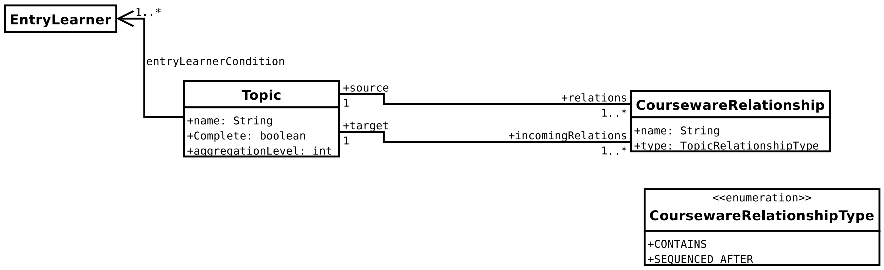

Fig. 1. CAVIAr courseware metamodel excerpt

4 Model Driven Engineering and Courseware Development

In our previous work we have outlined how Model Driven Engineering (MDE) methodologies, which are traditionally used in the development of software, can be used to develop courseware [10]. In this work the course creator defined a courseware sequence using an UML Activity Diagram, which was then transformed into a courseware specification using a model transformation language.

A metamodel defines the syntax and semantics of a model. Metamodels are defined by metametamodels. Model transformations have mapping defined at the metamodel level. Model transformations allow for the transformation of a model, which is an instance of one metamodel to a model which is an instance of a different metamodel. Figure 2, outlines model transformations defined at the metamodel level, and the actual mapping at the model level.

It should be noted that the metamodels and the model transformation definition must be defined as instances of a common metametamodel.

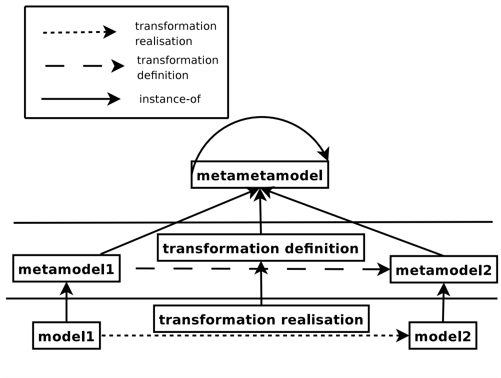

Fig. 2. Model transformations

Tool support for MDE is provided by the Eclipse Modeling Framework (EMF) [11]. We use EMF to represent CAVIAr models. EMF was used as it provided support for the following functions:

- a method for defining metamodels using ECore
- allowed for model transformations through the Atlas Transformation Language (ATL) [12]
- provided metamodel management infrastructure

5 Transforming LAOS to CAVIAr

In order to validate AEH defined by MOT using CAVIAr, the LAOS model must be used to generate a CAVIAr models. To do this we define metamodels for LAOS, one looking at LAOS static elements in CAF and the other its adaptive rules in LAG. We also define transformation from the LAOS metamodel to the CAVIAr metamodel, by identifying the relations between the metamodels.

In this section, we firstly outline the definition of CAF metamodel and its transformation relations to CAVIAr and then do the same for LAG. We note that the transformations specified here are example mappings, all model mappings can be customised by the course creator to represent their own opinions on the relationship between LAOS and CAVIAr.

5.1 CAF Transformation

To create a CAF ECore metamodel, we used the CAF XML definition, defined using a DTD [1]. This was converted to an XML schema using XMLSpy [13]. To create the CAF ECore Metamodel we converted the XML schema to an ECore model using EMF and then performed some minor alterations, as follows:

- created an explicit link between Link and Attribute
- added “value” attribute to CAF elements which contain text
- specified which relationships were ordered

The final CAF metamodel is illustrated in figure 3.

Once the CAF ECore metamodel is defined, the transformation between the CAF and CAVIAr metamodels can be defined using a model transformation language such as the Atlas Transformation Language (ATL) [12] or OMG’s Query View Transformation (QVT) [14]. Here we define the transformation specifications at a high level.

Generating the CAVIAr Domain Model In order to define the CAVIAr domain model, we have defined a model transformation from the LAOS domain model concept map to the CAVIAr domain model.

In this transformation, the CAF domain model concept is related to the CAVIAr domain model concept. The conceptual composition relationship in CAF, which relates two CAF concepts together, is transformed to the CAVIAr ConceptRelationship class of type “NARROWER”, where the contained concepts in LAG are narrower in scope to that of the containing concept in CAVIAr.

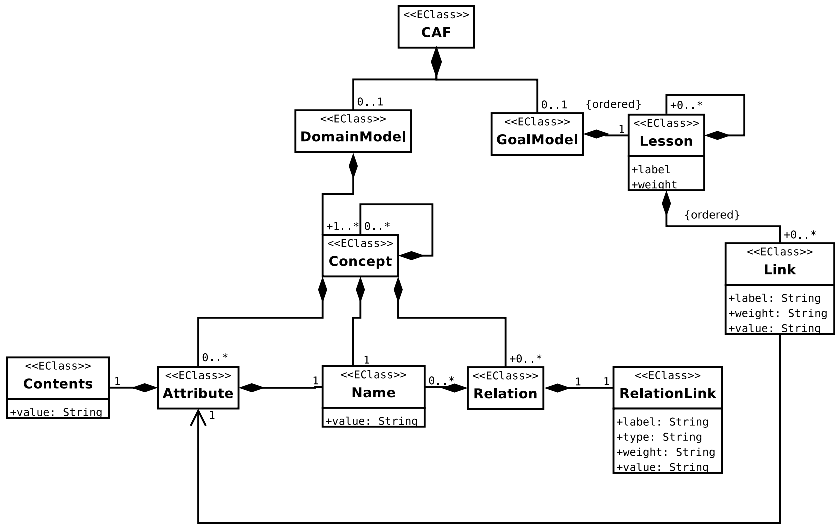

Fig. 3. CAF Metamodel defined using ECore

Generating the CAVIAr Learning Context Model The CAVIAr learning context is defined using the CAF goal and constraint model definition. Mapping is defined as follows:

– the CAF goal and constraints model is transformed into a single generic learner stereotype in CAVIAr
– CAF lesson goals are transformed into CAVIAr goals for the generic learner stereotype
– conceptual sequencing data in CAF lesson is transformed to PRE_REQUISITE relationships between concepts in CAVIAr

Generating the CAVIAr Courseware Model - Courseware Structure A courseware model is not defined by the CAF model, but can be derived using the domain model. In LAOS, the domain model contains the educational content to be delivered to the learner. We can therefore infer that each of the concepts in the domain model are also courseware topics in the courseware model.

In defining the transformation from the CAF model to the CAVIAr courseware model, we specify a 1:1 relation between the concepts in CAF and the CAVIAr courseware topics. Concepts contained in other concepts in CAF are transformed to subtopics in the CAVIAr courseware model.

Generating the CAVIAr Courseware Model - Learning Objects and Learning Object metadata In CAVIAr learning material is typically Learning Objects (LOs), and are annotated with metadata. This metadata can be used to determine the suitability of the LO at some point in the courseware. In AEH, the domain model defines what is in the AEH lesson. The domain model not only defines a conceptual structure of the AEH course but also defines the learning content. In the LAOS, the learning content is defined in concept attributes.

To generate LOs from the LAOS, we transform each conceptual attribute to a LO. The LO metadata is automatically derived for each LO generated, using the attribute type (e.g. title, conclusion) and the concept the attribute is associated with.

5.2 LAG Transformations

LAG rules are used to define adaptivity in LAOS (section 2). CAVIAr adaptation is provided by specifying restrictions on the sequencing of topics and restrictions on learner profiles which can access a topic. This type of adaptivity is defined using modeling constructs, such as defining a sequencing relationship between topics.

We wish to take the LAG adaptivity rules and transform them into CAVIAr courseware model restrictions. To do this the LAG language must be defined in the modeling technical space. We have defined a limited metamodel for the LAG abstract syntax in figure 4. This metamodel allows us to represent LAG in the modeling space by parsing a LAG rule and creating a LAG model. The LAG model can then be transformed and integrated into the CAVIAr model created using the CAF in section 5.1.

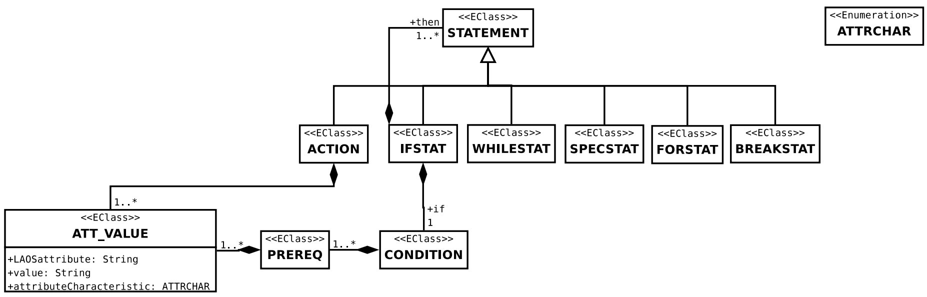

Fig. 4. LAG defined as ECore metamodel

Transformation rules can then be defined from the LAG metamodel to the CAVIAr metamodel. In the following we outline an adaptive rule which is commonly used in LAOS to define AEH, and describe the transformation definition which converts the LAG rule to the CAVIAr.

Transforming LAG Sequencing Rule LAG sequencing rules specify when a particular part of the domain model is accessed, it renders a different part of the AEH available to the learner.

An example of a LAG sequencing rule is as follows (listing 1.1):

Listing 1.1. LAG sequencing rule

```
IF (DM.Concept.title.access == 'true') THEN
   (DM.Concept.text.show == 'true')
```

The rule above states that if a domain model's concept title is accessed, then the text for that domain model concept is shown to the learner. This type of LAG rule is made up of two different parts, an IF condition and an action. The condition and action are composed by checking (condition) and then setting (action) a characteristic of a domain model concept's attribute in LAOS. The condition checks the attribute "title" for domain model concepts has been accessed - "access" being the characteristic. In turn, the action sets the LAOS "text" attribute to be shown - "show" being the characteristic being set.

This rule is parsed and creates an instance of the the LAG metamodel - a LAG model - as illustrated in figure 5.

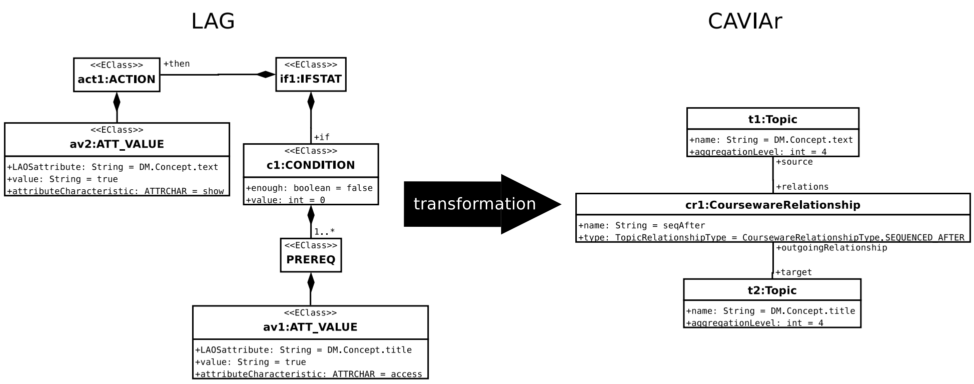

Fig. 5. Transformation of LAG model to CAVIAr courseware model

When a LAG model has been constructed for the rule in listing 1.1, the LAG rule can be transformed into the CAVIAr courseware model. To do this a transformation mapping from the LAG metamodel to the CAVIAr metamodel is defined. This transformation states when DM.Concept.title attribute is accessed show the DM.Concept.text attribute. The transformation maps this type of LAG rule to a CAVIAr courseware model where each attribute in the LAG condition and action is a courseware topic. The topic mapped to the title attribute is the source of a "SEQUENCED_AFTER" CoursewareRelationship where the target is the topic mapped to the text attribute. We have demonstrated this through an example transformation in figure 5.

6 Validating AEH using CAVIAr

In this section, we generalise the LAOS validation methodology we have presented in this paper and examine how AEH is validated in general.

When AEH is being validated for the first time, a metamodel for the AEH data models must firstly be defined. This allows the AEH to be used in the modelling technical space. The AEH native data models must be parsed to create an instance of the metamodel defined. Transformations to the CAVIAr must then be defined to map the AEH being used to the CAVIAr. The AEH metamodel and transformation to CAVIAr once defined, can be reused.

To validate AEH the course creator must then define the validation model for the CAVIAr. The validation model specifies constraints that must be adhered to in the AEH and are defined in the context of the CAVIAr models. For example, the course creator may specify that all concepts covered in the AEH are introduced with a motivating example and delivered before any other material on that concept is delivered to the learner. The course creator may feel that the AEH has been defined this way but wants to guarantee it through CAVIAr validation. The course creator defines this as a constraint on the CAVIAr model.

The validation is then run using the CAVIAr validation engine, this validates the generated models against the CAVIAr model constraints specified in the validation model. If any of the validation model constraints are breached, the course creator is notified and he or she can then rectify them in the AEH.

7 Conclusion

In this paper we have described courseware validation as a method for course creators to minimise the risk involved in creating and deploying AEH. The CAVIAr has been introduced in the AEH context, as a way for course creators to test the AEH developed for specific pedagogical concerns.

To enable interoperability between the LAOS and CAVIAr, we have outlined the application of MDE technologies and methodologies, provided model mapping from the LAOS to the CAVIAr, and detailed an implementation infrastructure with which the conversion from LAOS to CAVIAr can take place. AEH interoperability has been investigated in a number of papers, much of this work concentrates on once off conversions between two AEH technologies [7, 1, 15, 16]. In this paper we have outlined how MDE offers a generic approach to AEH interoperability, where interoperability can be achieved when a metamodel is defined for the AEH technology in use and transformations between the metamodels are implemented. The methodology outlined is also highly customisable, all the AEH to CAVIAr mappings can be changed to reflect the course creators own opinions on metamodel relationships.

References

1. Cristea, A., Smits, D., deBra, P.: Towards a generic adaptive hypermedia platform: a conversion case study. Journal of Digital Information **8**(3) (2007)

2. Dagger, D.: Personalised eLearning Development Environments. PhD thesis, University of Dublin (2006)
3. Melia, M., Pahl, C.: An information architecture for validating courseware. In Massart, D., Colin, J.N., eds.: Proceeding of the First International Workshop on Learning Object Discovery and Exchange (LODE2007) at EC-TEL2007, Crete, Greece, CEUR Workshop Proceedings (2007)
4. Cristea, A.I., de Mooij, A.: LAOS: Layered WWW AHS Authoring Model and their corresponding Algebraic Operators. In: Proceedings of The Twelfth International World Wide Web Conference (WWW03), Alternate Track on Education, ACM (2003)
5. Cristea, A.I., Verschoor, M.: The LAG Grammar for Authoring the Adaptive Web . In: Proceedings of The International Conference on Information Technology: Coding and Computing (ITCC'04) Volume 1. (2004) 382–386
6. Cristea, A.I., Smits, D., de Bra, P.: Writing MOT, Reading AHA! - converting between an authoring and a delivery system for adaptive educational hypermedia. In: Proceedings of The Third International Workshop on Authoring of Adaptive and Adaptable Educational Hypermedia at AIED05. (2003)
7. Cristea, A.I., Stewart, C., Brailsford, T., Cristea, P.: Evaluation of Interoperability of Adaptive Hypermedia Systems: testing the MOT to WHURLE conversion in a classroom setting. In: Proceedings of The Third International Workshop on Authoring of Adaptive and Adaptable Educational Hypermedia at AIED05. (2003)
8. Melia, M., Pahl, C.: Pedagogical validation of courseware. In Duval, E., Klamma, R., Wolpers, M., eds.: Proceedings of the Second European Conference on Technology Enhanced Learning. Number 4753 in LNCS, Springer (2007)
9. Ullrich, C.: Course generation based on HTN planning. In: Proceeding of the Thirteenth Annual Workshop of the SIG Adaptivity and User Modeling in Interactive Systems. (2005) 74–79
10. Melia, M., Barrett, R., Pahl, C.: A Model-Based Approach to SCORM Sequencing. In: Proceeding of the Sixth Annual Irish Educational Technology User's Conference (EdTech06) - Research Track, ILTA (2006)
11. Budinsky, F., Brodsky, S.A., Merks, E.: Eclipse Modeling Framework. Pearson Education (2003)
12. Jouault, F., Kurtev, I.: Transforming Models with ATL. In: Proceedings of the Model Transformations in Practice Workshop at MoDELS 2005. (2005)
13. Altova: Altova Xmlspy 2005 User & Reference Manual. Vervante (2005)
14. OMG: Meta Object Facility (MOF) 2.0 Query View Transformation (QVT) (2005) OMG Final Adopted Specification.
15. DeBra, P., Santic, T., Brusilovsky, P.: AHA! meets Interbook, and more... In Richards, G., ed.: Proceedings of the AACE ELearn Conference 2003, AACE (2003) 57–64
16. Power, G., H.C., D., Cristea, A., Stewart, C., Ashman, H.: Goal oriented personalisation with scorm. In: Proceedings of the Fifth International Conference on Advanced Learning Technologies (ICALT2005), IEEE Computer Society (2005) 467– 471

An adaptive hierarchical questionnaire based on the Index of Learning Styles

Alvaro Ortigosa, Pedro Paredes, Pilar Rodriguez

Escuela Politécnica Superior, Universidad Autónoma de Madrid
28049 Madrid, Spain
{alvaro.ortigosa, pedro.paredes,
pilar.rodriguez}@uam.es

Abstract. One of the main concerns when providing learning style adaptation in Adaptive Educational Hypermedia Systems is the number of questions the student has to answer. With respect to learning styles, it is possible to decrease the number of versions taking into account the general tendency of the student and not the specific score obtained in each dimension. In this paper we present a new approach to reduce the number of questions of Index of Learning Styles (ILS) questionnaire based on Felder-Silverman's Learning Style Model (FSLSM). The results obtained in a case study with 330 students are very promising. It was possible to predict students' learning styles with high accuracy and only a few questions.

1 Introduction

In order to provide adaptation, Adaptive Hypermedia Systems (AHSs) [3] need to store and maintain information about the user, which constitutes the user model [9]. Building user models implies gathering information about the users and transferring this information into the model. Many systems use questionnaires for detecting users' features while others try to infer them from user interactions with the system.

In the area of Adaptive Educational Hypermedia (AHE), one of the student features frequently used with adaptation purposes is learning style. In recognition of the fact that individuals learn in different ways, a body of research and techniques has been developed, which attempts to categorize individual variations while satisfying different learning style preferences.

Felder and Silverman created a learning style model (FSLSM) [4] that has been widely used in technology-enhanced learning. It describes learning styles distinguishing between preferences on four dimensions (active/reflective, sensing/intuitive, visual/verbal, and sequential/global). Information about these preferences can be extracted from the corresponding questionnaire (ILS) [5], which contains 44 questions. We have used FSLSM and ILS in previous works [1] [10] [12].

Even when information about learning styles is very useful for adapting the education material to each student, answering the 44 questions from the ILS is a time con-

[1] This work is supported by the Spanish Ministry of Education and Science, TIN2007-64718.

suming and boring task. That is especially relevant if the system requires more information from the students, beside the learning style. In this paper we propose an approach to reduce the number of questions needed to determine the learning style of each student.

Next section describes the goal of the work in detail. Section 3 and 4 explain how the study was developed, while section 5 presents the results. Section 6 describes some related work and finally section 7 presents the conclusions.

2 The goal

Most of the current AEHSs that provide adaptation based on learning styles use the ILS questionnaire to obtain the learning style model of each student. ILS produces information about 4 dimensions of learning styles, using 11 questions for each dimension. The score are obtained by subtracting the number of answers related to one category from the number of answers related to the opposite category. In this way, the final results from the test are four scores (odd numbers ranging from -11 to 11), one for each dimension. That is, there are 12 possible different results for each one.

This information provides many opportunities for adaptation, because an AEHS could deliver 12 different versions of the educational material considering only one dimension of the learning style model. On the negative side, adapting to learning styles requires the student to answer 44 questions about his/her preferences, which many times it is considered a heavy additional burden.

However, most of the times there is not a different version of the course for every possible value of the questionnaire, but students are clustered in classes covering different values. For example, Felder et al. [5] recommend grouping the students into five categories for every dimension. If a student gets a score from 1 to 3 in any dimension, he/she has a mild preference but his/her learning style is well balanced. Differently, if the score is from 5 to 7, the student has a moderate preference and he/she will learn more easily in teaching systems that favor that dimension. Finally, if the student scores from 9 to 11, he/she could have difficulties when learning through a system that does not support this preference.

In previous experiences with adaptive courses, we have found that authors use to prefer to classify the students into three categories for every dimension: **low**, **neutral** and **high**. In this case, students having, for example, values between -11 and -5 in a given dimension would be provided with the same version of the adaptive course, students with values between -3 and 3 would receive a second version and students having between 5 and 11 would receive a third one.

In this context, the system only needs to know the class of a given student for every dimension, but not the exact value. As a consequence, it does not need to ask the student the 11 questions of the ILS, but only enough questions to discriminate his/her class. The problem is: which questions (among the 11) would provide enough information about the student learning style?

This problem is a variation of the general question approached by the Item Response Theory (ITR) [13]. ITR mostly focuses on the problem of analyzing the power of a question or a whole test to evaluate, for example, knowledge or IQ of a person.

Our goal is to provide AEHSs with the ability to classify the learning style of a given student with so few questions as possible. Eventually, we seek to obtain an algorithm capable of asking different questions to different students: the next question to be posed is calculated considering the answers given so far by the student (figure 1). It is important to highlight that we do not attempt to propose new questions for finding the student learning style, but only to select the more relevant ones **for each student** from the ILS.

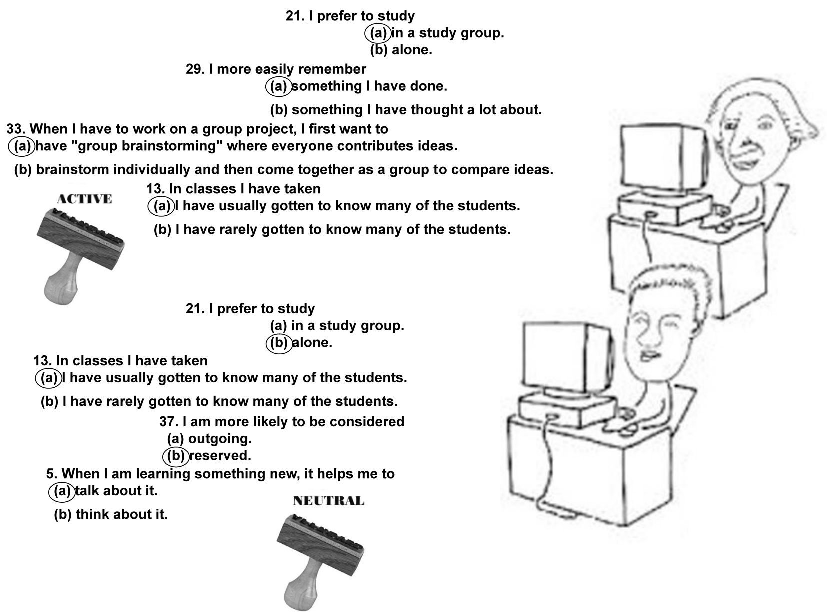

Fig. 1. Different questions for different students

Classification is one of the main goals of data mining techniques [15]. In general, these techniques learn a classification model from the observation of (already classified) instances. Once the model has been learnt, it can be used to classify new instances whose class is unknown.

This work shows how classification techniques can be used to learn which questions should be asked to each student in order to reduce the number of answers needed to classify his/her learning style.

3 Data collection

Data mining techniques are based on the analysis of samples in order to find patterns in the data; this knowledge can be used to classify new examples, considering the class of similar patterns in the sample.

Samples of students belonging to three different populations were used to generate the results presented on this work.

- Sample 1: 42 students from Secondary School level (IES "Agora", Madrid).
- Sample 2: 80 students from a Vocational School (post-secondary level, CIFP "Jose Luis Garci", Madrid). They were studying audio-visual technology.
- Sample 3: 200 students from the Computer Science and Engineering degree at the Universidad Autónoma de Madrid.

As a result, the study is based on the answers to the ILS questionnaire from 330 students who were between 15 and 30 years old. In the rest of the paper, the term "sample" will make reference to the whole set of students, considering the aggregation of the three samples described above.

Figure 2 shows the frequency of each ILS dimension for the sample. Dim1 to dim4 correspond to the Active/reflective, Sensing/intuitive, Visual/verbal and Sequential/global dimensions, respectively. These frequencies do not follow the normal distribution, but fortunately this is not a requirement of the techniques used to analyze the data. Not surprisingly, data distributions fairly accurate to the distributions found on a previous experiment with similar population [1]. Regarding the distribution between genders, 101 were women and 229 were men.

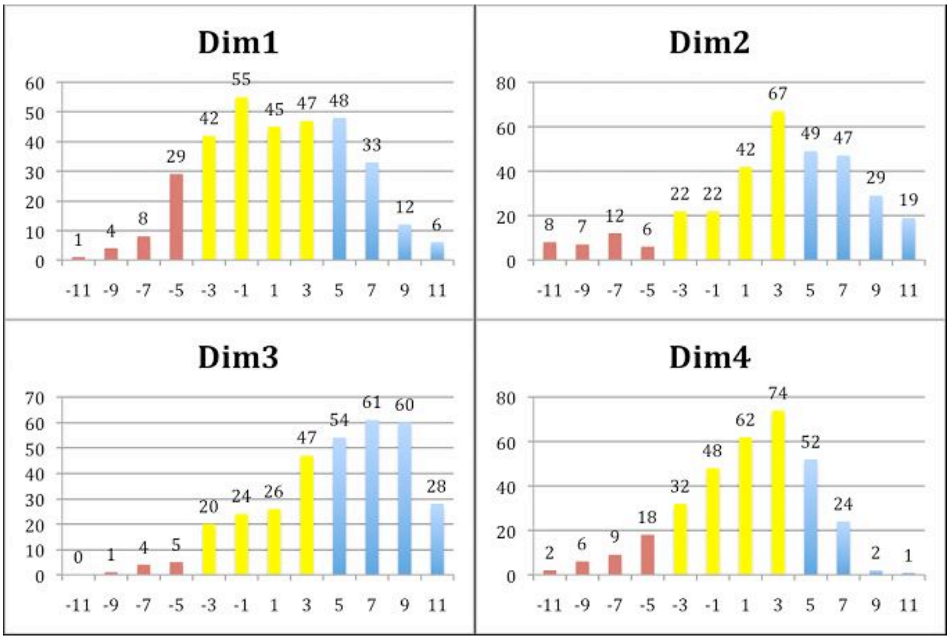

Fig. 2. Distribution for every dimension

4 Methods

The data were processed and the students divided into three classes: **high** (from 11 to 5), **neutral** (from 3 to -3), and **low** (from -5 to -11). This dataset was analysed using the *Weka* workbench of data mining algorithms [15].

Classification algorithms learn a model based on the instances of the dataset, where each instance is described as a collection of attributes. In this case, an instance or example was formed by the data of a given student: the answer (a or b) he/she gave to each of the ILS questions and the class assigned to each leaning style dimension.

Considering the goal of this work, decision trees are very convenient tools. Nodes in a decision tree involve testing a particular attribute of the instance to be classified. Depending on the attribute value, the corresponding descendent branch is followed. This procedure is recursively applied until a leaf is reached. Usually, each leaf has a label with the class to be assigned to the instances that reach that leaf. As a consequence, along each path from the root to a leaf they can be used, potentially, different attributes from the instance to be classified.

Decisions trees have two properties that are well suited for the goal of this work:

- When building decision trees, the criterion for choosing the next attribute to be used to split the data is to maximize the information gain. In other words, they select the most relevant attribute for a given subset of the sample.
- Decision trees provide an explicit representation of the classification model, enabling the construction of dynamic tests based on the attributes (questions) used by the tree.

Particularly, on this work a variation of the C4.5 algorithm [11], called J4.8 [15], for building decision trees was used. In order to avoid overfitting, 10 folds cross-validation was used. This is very important; because the goal was to get the relevant questions, the pruning of the resulting trees was reduced.

5 Results

Table 1 shows the average path from the root to the leaves in the classification tree for each dimension, considering the number of training examples that reached each leaf. That is, each length represents the expected number of questions the AEHS should ask before being able to classify a student for that dimension.

Table 1. Estimated number of questions for each LS dimension.

	Active/Reflective	Sensing/Intuitive	Visual/Verbal	Sequential/Global
Questions	4.97	4.06	4.96	4.28

The questions posed to each student are selected on the fly, accordingly to the classification tree generated for each dimension. Figure 3 shows, for example, the classification tree generated by the J4.8 algorithm from the sample for the *Sensing/intuitive* dimension.

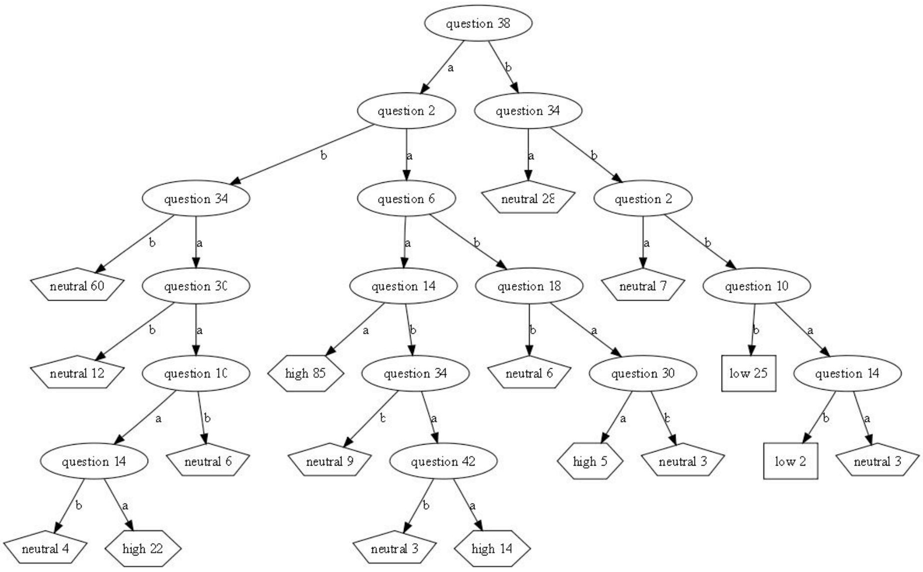

Fig. 3. Decision tree for the *Sensing/intuitive* dimension.

Even if all the examples on the sample are well classified, the classifier is assumed to make some mistakes when classifying new instances (students). In order to estimate the predicted classification error, ten fold cross-validation was used [15]. Table 2 shows the estimated prediction error for each dimension.

	Active/Reflective	Sensing/Intuitive	Visual/Verbal	Sequential/Global
Error (%)	4.29	1.36	2.92	3.41

Table 2. Estimated classification error for each LS dimension.

It should be noted that classification mistakes happen with students having values on the border between classes. For example, sometimes a student with a value -5 in a given dimension is wrongly assigned to the *neutral* category, instead of *low*. This type of mistakes is not severe, because a student with a value -5 will probably be well assisted by a "neutral" version of the educational material.

During the data analysis it was also observed that training the classification trees with less examples produced both larger errors and longer paths from the root to the leaves. Even though it is possible that significant larger samples would produce shorter trees with the same level of confidence, the tests developed do not seem to indicate that.

It is also interesting to describe the results when the three original samples were individually analyzed. Even if the expected error increased, the learning algorithm mostly selected the same attributes (questions) for the higher portions of the trees. This fact indicates two things:

a) The relevance of a question does not vary significantly with the age of the student.

b) The trees seem to converge to a common tree, independently from the origin of the sample, or at least to a common subset of questions.

6 Related work

The use of questionnaires, although usually provides accurate information, can be very time-consuming. Some works have investigated the use of Bayesian networks [6], behavior patterns [7], user-mouse interaction [2], and feed-forward neural networks [12] to detect learning styles starting from information of user behavior in educational websites (tasks done, time spent, scores obtained). However not all characteristic behavior described in the learning style model can be mapped and identified from the behavior in a specific learning system.

A previous work [8] tried to identify the five most representative questions for each dimension of the ILS according to frequencies analysis. Nevertheless they investigate the relationship between these questions and semantic groups established by them instead of trying to reduce the number of questions of ILS. Comparing their ranking and our decision trees we can see that those relevant questions in [8] are in the four highest levels of the trees.

7 Discussion

In this work we have presented a new approach to predict students' learning styles that reduces the number of questions of ILS questionnaire that each student has to answer.

The results of the case study show that some questions from the ILS are more relevant that others, in the sense that they provide more information about the general tendency of the student along the corresponding dimension. Particularly, using a sample with 330 students, we were able to build classification trees that need, on the average, between 4 and 5 questions to classify a learning style dimension for each student. These results are very promising since the prediction accuracy obtained is very high (between 1.36 and 4.29% depending on the dimension).

Even if different samples could produce different classification trees, considering that each tree is concerned only with 11 questions, the size of the sample is enough to consider that classification trees would not be much different for other samples. Actually, the three individual samples show very little difference of distribution among them and the classifications trees built for individual samples tend to use the same discriminating questions.

Even so, an author intending to use the best possible sequence of questions to classify students' learning style could build classification trees based on samples from her target population. However, if the sample is not large enough, these "specific" trees would produce more errors than generic ones.

A possible bias of the studied sample is the proportion of men to women (more than 2 to 1). However, results from the case study show no significant difference between the results of the ILS for men and women. Moreover, none of the classification trees used the gender as a discerning attribute. In other words, knowing the gender of a student does not provide information about his/her learning style.

It is also possible to create more classes for each dimension, for example the five categories proposed by Felder and Silverman. However, it would be needed to ask more questions in order to refine the classification. Besides, additional example instances would be needed in order to reach good precision levels with more classes.

We plan to extend our study collecting and analyzing data from different groups of students. In addition, we plan to eventually combine information extracted from ILS questionnaire with that related to the type of information selected, activities done, time spent on each one, mouse movements and so on.

References

1. Alfonseca E., Carro R.M., Martín E., Ortigosa A., Paredes P.: The Impact of Learning Styles on Student Grouping for Collaborative Learning: A Case Study. User Modeling and User Adapted Interaction, special issue on "User Modeling to Support Groups, Communities and Collaboration", 16, 3-4, 377–401 (2006)
2. Bergasa-Suso J., Sanders D.A.: Intelligent browser-based systems to assist Internet users. IEEE Transactions on Education, 48, 4, 580-585 (2005)
3. Brusilovsky, P.: Adaptive hypermedia. User Modeling and User Adapted Interaction, Ten Year Anniversary Issue (Alfred Kobsa, ed.) 11 (1/2), 87-110 (2001)
4. Felder R.M., Silvermann L.K.: Learning and teaching styles in engineering education. Journal of Engineering Education 78, 674–681 (1988)
5. Felder R.M., Soloman, B.A.: Index of Learning Styles Questionnaire. Available online at: http://www.engr.ncsu.edu/learningstyles/ilsweb.html
6. García P., Schiaffino S., Amandi A.: An enhanced Bayesian model to detect students' learning styles in Web-based courses. Journal of Computer Assisted Learning (OnlineEarly Articles). doi:10.1111/j.1365-2729.2007.00262.x.
7. Graf S., Kinshuk: An Approach for Detecting Learning Styles in Learning Management Systems. Sixth International Conference on Advanced Learning Technologies (ICALT), 161-163 (2006)
8. Graf S., Kinshuk, Leo T., Viola S.R.: In-Depth Analysis of the Felder-Silverman Learning Style Dimensions. Journal of Research on Technology in Education, 40(1), 79-93 (2007)
9. Kobsa A.: Generic User Modeling Systems. User Modeling and User-Adapted Interaction: The Journal of Personalization Research 11, 49-63 (2001)
10. Paredes P, Rodríguez P.: A Mixed Approach to Modelling Learning Styles in Adaptive Educational Hypermedia. Advanced Technology for Learning, ACTA Press 1(4), 210–215 (2004)
11. Quinlan, J.R.: Improved Use of Continuous Attributes in C4.5, Journal of AI Research 4, Morgan Kaufmann, 77-90 (1996)
12. Sánchez Hórreo V., Carro R.M: Studying the Impact of Personality and Group Formation on Learner Performance. Groupware: Design, Implementation, and Use. LNCS 4715, Springer-Verlag, 287–294 (2007)
13. Van der Linden, W.J., Hambleton, R.K.: Handbook of modern item response theory. New York: Springer (1997)
14. Villaverde J., Godoy D., Amandi A.: Learning styles' recognition in e-learning environments with feed-forward neural networks. Journal of Computer Assisted Learning 22, 197–206 (2006).

15. Witten I., Frank E. Data Mining – Practical Machine Learning Tools and Techniques. Elsevier. (2005).

Collaborative Adaptation Authoring and Social Annotation in MOT

Fawaz Ghali[1], Alexandra I. Cristea[1], Craig Stewart[2] and Maurice Hendrix[1]

[1] Department of Computer Science, University of Warwick,
Coventry, CV4 7AL, United Kingdom
{F.Ghali, A.I.Cristea, M.Hendrix}@warwick.ac.uk
[2] Department of Electronic Engineering, Queen Mary University of London,
Mile End Road, London, E1 4NS, United Kingdom
Craig.Stewart@elec.qmul.ac.uk

Abstract. The Web is constantly evolving, in fundamental and innovative ways. Social annotation and collaborative authoring facilitate and change the process of creating and sharing information. In this paper, we propose a new design of the authoring system MOT (*My Online Teacher*), focusing on *collaborative* authoring and *social* annotation. The goal behind this is to define improved adaptive materials based on personalization and recommendation. We start the collaboration design process by discussing its features, characteristics and creating a survey with a group of third-year students in a "Web Programming" course at the University "Politehnica" of Bucharest. Results confirm a consistent association between the authors and the students in the proposed prototype; the key elements of the collaboration are rating and tagging the attributes, in addition to feedback the content in the *domain model*; moreover, the privileges are defined at the level of the lesson and the link in the *goal model*.

Keywords: Collaboration design, social annotations, collaborative authoring, Web 2.0, adaptive hypermedia, MOT.

1 Introduction

Collaborative authoring and social annotation are two faces of same coin: both rely on cooperation, but in different ways. Whilst *collaborative authoring* (annotation and editing during writing [8]) creates/modifies the actual web resources, *social annotation* (or annotation during reading [8]) facilitates the adding/editing/modifying of information in a web resource, without changing the resource itself [10]. The main goal for defining cooperation (collaborative authoring and social annotation) in MOT [3] is to allow multiple authors to contribute in the authoring process. Thus, the authored materials foster a new level of knowledge (both of creation and of use) by aggregating information from many users. In principle, the more users that contribute to the authoring process, the more valuable the final *stable* material is. Stability is important to reach, as systems such as Wikipedia show, because only then a consensus of the community is certain. Whilst a resource is still changing, its value is less certain to that community.

2 MOT and CAF

MOT [3] is an authoring tool that can be used for authoring adaptive hypermedia courses. MOT is based on the LAOS (*Layered WWW AHS Authoring Model and their corresponding Algebraic Operators*) framework [4], and consists of: 1) *Domain layer*: defining the conceptual domain model (DM), built of atomic and composite concepts, where each concept has a set of attributes. 2) *Lesson layer*: defines multiple flexible lessons from a given domain map or combination of domain maps. 3) *User - adaptation - and presentation layer*: described by adaptation strategies. In MOT, the contents (Conceptual and Lesson layers) can be exported into Common Adaptation Format (CAF) xml files [5] (Figure 1).

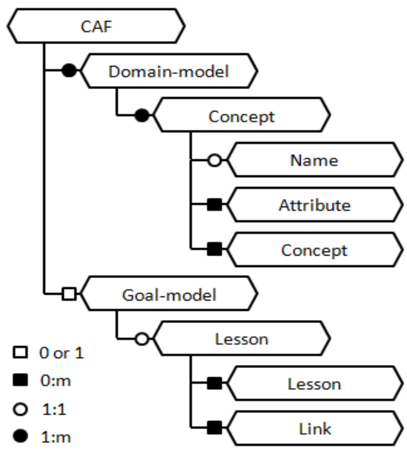

Fig 1. CAF XML structure; where ‘x:y’ means ‘the range x to y’; ‘m’ stands for ‘more’.

CAF represents the actual data structure of the domain and lesson part of the MOT database by using XML, which is more suitable for conversions. A CAF XML file has: 1) a *Domain model*: consisting of one or more domain maps, each with a set of concept(s); and each concept with a set of attribute(s) that describes related domain data (and link to the actual content). A concept may have sub-concept(s) and associations to other concepts. An attribute has a name and contents. 2) a *Goal model*: consisting of a single goal map, representing the actual lesson, which may have a set of sub-lesson(s). Each lesson has a set of link(s), where each link points to an attribute in the domain model. The link has two attributes: weight and label, which are used to determine the adaptive requirements via adaptation strategies.

3 Extending Collaboration in MOT

MOT is an adaptation authoring tool, which already supports some basic collaboration activities such as: 1). General access (visualization) to other author's domain maps, as well as lessons. 2). Keyword-based access (visualization) of existing domain concepts, created by the current author or by other authors. 3). The possibility of copying a domain concept across from another of the author's own domain map(s), which allows reuse of previously created materials. 4). The possibility of linking to concepts from someone else's domain map(s), which corresponds to referring in one's own book to someone else's book (e.g., adding a full quote). 5). Semi-automatic search and linking function (with weights and labels) to link domain concepts from any domain map, be they authored by the same author or not, to another (related) domain concept, which helps the author in finding related domain concepts to the one they are currently authoring, so they can reuse material, or refer to it, as necessary. 6). The possibility of creating a lesson based on someone else's domain map(s): this is similar to creating a lesson based on someone else's book(s). 7). The possibility of creating a lesson including other lesson(s) created by other authors, which corresponds to reusing the lesson materials of other teachers on the web, as long as they post it and allow such reuse.

On top of the above, the design, as initially proposed in this paper, is focusing on utilizing two aspects:

1) *Collaborative authoring*: where multiple authors can contribute in the authoring process. The question here is at what level in the CAF structure as shown in Figure 1 this collaboration should be. Possible options are, e.g., collaboration at the level of a domain concept, domain attribute, lesson, link, whole domain map, whole lesson map, etc. E.g., collaboration at the level of a concept in the CAF file would allow a user to edit previous concepts or create new ones (and/or sub concepts).

2) *Social annotations*: where multiple users (authors and/or students) can annotate the content of the attribute (tag/rate/feedback) and share this annotation with other users. A similar question appears as in the collaboration: which level of granularity is needed for social annotations? For instance, students rating at the level of the attribute in the CAF file would mean that an author could get feedback of the usefulness of each of the attributes he/she created.

Both collaborative authoring and social annotation will require the introduction of a more refined system of authorization (level of privileges). The question here is at which level of granularity privileges should be defined at. For instance, if privileges for authoring can be only at the level of whole domain maps, then it means that different authors have full editing rights over the whole contents of a domain map. If, however, such privilege is granted at a lower level of granularity, e.g., at the level of a link or lesson in CAF, then this would correspond to one domain attribute. In such a way, for each domain attribute, different accesses can be set. E.g., user A can grant user B editing rights over the 'keyword' attribute of a specific concept, say 'German pronouns', but not over the 'text' attribute, which contains the main text of this concept. User A could however at the same time grant editing rights to user C for the 'text' attribute, as well as, for instance, the 'keyword' attribute – but not over the 'video' attribute. This allows for an appropriate use of specialists, who can receive specific rights for the concepts and attributes they are specializing on, and nothing else. Thus, it seems appropriate to define users and groups at the level of links and lesson in the CAF file (as explained in detail in section 5). These design ideas and initial mental speculations were further transformed into design hypotheses and then tested, as explained further in the paper

4 Hypotheses of the Collaboration MOT System Extensions

The study of related research, as shown in section 6, generated a number of design hypotheses on social annotation, collaboration and roles, and the required granularity:
H1. Adding collaborative facilities via social annotation (rating/feedback/tags – e.g., keywords) is useful for both authors and students.
H2. The tools to realize collaborative authoring should be based on a combination of semantic web (ontology-based structures) and social web techniques (Web 2.0: 'free' tags and annotation).
H3. Specifically, social annotation should be performed at the level of domain concept attributes in MOT.
H4. Social collaborative tools should support both author-author and author-student collaboration.
H5. Collaboration is needed at the level of both users and groups.
H6. Users and group privileges should be defined at the level of links or lessons.
Thirty 3rd year students contributed to this design stage of Collaboration MOT, by answering a questionnaire and reflecting refined hypotheses.

4.1 Testing the Hypotheses

We prepared a questionnaire (see the ANNEX) based on our hypotheses, in which we asked eight questions about the design of Collaboration MOT. A group of thirty students studying in a "Web Programming" course, partially delivered via distance learning, collaborated in the creation of new content in MOT and answered our questionnaire. The students were enrolled in the 3rd year of Computer Science at the Politehnica University of Bucharest, Romania. As their own course is partially delivered online, students can be expected to act as social annotators, and also to participate in the collaborative authoring process. Thus, the study performed can be considered real inquiry into what users need (e.g. supporting them in accomplishing a particular task, such as creating a part of a course description). Moreover, by being computer science students, the chosen group could be considered computer and software applications savvy and thus able to have the expertise or foresight to choose software features appropriately. Before the students had to fill the questionnaire, they were made familiar, via lectures, with Semantic Web concepts and technology (e.g., XML, XPath, RDF), Social Web concepts and technology (folksonomies, Web 2.0 concepts, etc.), and via hands-on experiments, with authoring environments (MOT) and learning environments (Sakai , AHA!). The questions that they were asked focused on our two principles of collaboration: collaborative authoring and social annotation.

Summarizing the results, we can say that 15 out of 30 respondents believed that collaboration should use both Web 2.0 as well as Semantic Web (Ontology/RDF/OWL) techniques; 26 respondents believed authoring should use social annotation techniques (such as tagging, rating, and feedback mechanisms); 24 answered that collaboration (social annotation and collaborative authoring) should be done by students and authors together; 21 responded that collaboration must be defined using users and groups; whereas 10 thought that the privileges should be defined at the level of lesson as well as with links to the goal model.

We applied a Chi-square test to verify if our observations match our hypotheses. We chose the chi-square test because our questionnaire used categorical data. The degrees of freedom associated with our data is calculated as follows: Df = number of categories – 1 As shown in Table 1, all results are statistically significant, as tested with the help of the Chi-Square test (with significance level $p<= 0.05$).

Table 1. Questionnaire statistics

Question	Chi-Square	Df	p	Hypotheses
Q1	11.862	2	.003	H2
Q2	41.448	2	.000	H1
Q3	26.034	3	.000	H3
Q4	24.069	2	.000	H4
Q5	12.448	1	.001	H4
Q6	19.931	2	.000	H5
Q7	11.345	5	.047	H6
Q8	11.345	5	.047	H6

Thus, our hypotheses are confirmed (Table 1), and we conclude that the groups of users of MOT would like to see the type of collaboration as described by our hypotheses. We also performed some more detailed analyses of the data. For example, for question 1, concerning the best application of collaborative authoring in MOT, the four answers (social web, semantic web, both, none), are selected with frequencies (14, 1, 15, 0), respectively. Beside the clear preference of both social and semantic web application together ($p=.003 <.05$, based on Chi-Square), we can also pairwise compare the two answers, separately. We notice a clear preference of social web techniques ($p=.049<.05$ as per a Binomial Test).

5 Design of Collaborative Authoring and Social Annotation

The proposed inclusion of collaborative authoring in MOT is focused on allowing groups of authors to contribute in the authoring process by providing editing facilities at the level of domain model concepts. After authoring the course, we propose another level of collaboration among students/authors, to interact with the authored material, by utilizing social annotation techniques, such as tagging, rating, and feedback. Thus, collaboration can be presented as collaborative authoring (create/edit) which can occur at the creation stage of the authored material (authors only), or at the usage stage (authors and students) or as social annotation (describe/evaluate/opinion).

The new Collaboration MOT DTD of the CAF file is extended, as shown below (extensions shown in bold). The domain concept is extended, allowing an arbitrary number of tags. For backwards compatibility purposes, zero tags are also allowed.

```
<!ELEMENT concept  (name, attribute*, concept*, tag*)>
```

The attribute is also extended, to allow for evaluation, and a list of tags and opinions. Evaluation can utilize a rating or voting system, whereas the opinion refers to the feedback or the comment created/edited by the annotator (student and/or author).

```
<!ELEMENT attribute (name, contents, tag*, evaluation?
opinion*)>
<!ELEMENT tag (user, keyword) >
<!ELEMENT opinion (user, feedback)>
```

The lesson element is extended to allow annotation by individual users that access this lesson, as well as by groups of users. Both authors and students can annotate the lesson. Despite the fact that user annotations appear in the DM, the privileges of the users (as well as the groups) are defined in the GM (at the level of the links and lessons), because the GM is used to point to the actual domain content (concepts), it is therefore more appropriate to define privileges at this level.

```
<!ELEMENT lesson  (link*, lesson*, user*, group*)>
```

The elements of the tagging mechanism are shown below, and are also added to the collaboration DTD. Users can tag data with their own keywords, evaluate resources, and give feedback.

```
<!ELEMENT keyword (#PCDATA)>
<!ELEMENT evaluation (#PCDATA)>
<!ELEMENT feedback (#PCDATA)>
```

The new DTD defines privileges based on individuals or groups:

```
<!ELEMENT user (#PCDATA)>
<!ELEMENT group (#PCDATA)>
```

The use of this new DTD extension is shown below, where one of the content concepts and attributes is commented upon and evaluated by students:

```
<CAF>
 <domainmodel>
  <concept>
    <name>Collaboration</name>
    <tag>
     <user>Jessica</user>
     <keyword>Social annotation</keyword>
    </tag>
    <attribute>
     <name>Introduction</name>
     <tag>
       <user>Rachel</user>
       <keyword> Relative </keyword>
     </tag>
     <evaluation>80%</evaluation>
     <opinion>
       <user> Jessica </user>
       < feedback >
            I understood it.
        </ feedback >
     </ opinion >
     < opinion >
```

```
        <user>Rachel</user>
        <feedback>
              Excellent work.
          </feedback>
      </ opinion >
      <contents> Information about collaboration.<contents>
     </attribute>
   </concept>
  </domainmodel>
  <goalmodel>
   <lesson weight= "" label= "beginners" >
     <user>Jessica</user>
     <group>Group 1</group>
     <link weight= "" label= "beginners" >
        Collaboration\Introduction
     </link>
   </lesson>
  </goalmodel>
</CAF>
```

5.1 Screenshots of Collaborative MOT

The general overview of the system is shown in Figure 2. *Collaborative MOT* will display the concept at the top, tags under the concepts, and the rating of the concept under the tags. Feedback from the users will be displayed at the bottom.

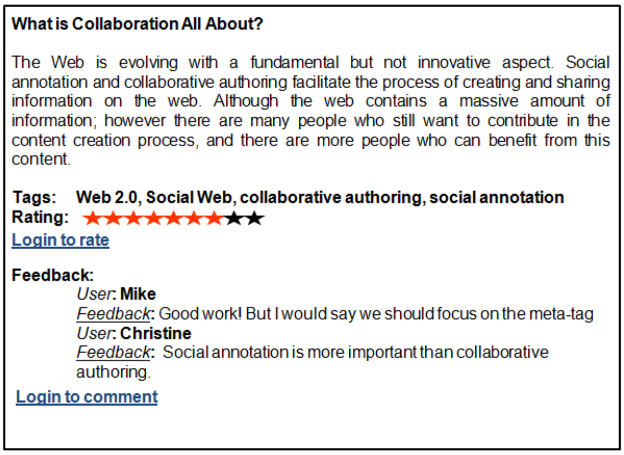

Fig 2. Social annotation in MOT

The author screen is shown in Figure 3. When the author logs into the system, a set of collaborative options will be displayed (add a concept, add a sub-concept, add an attribute), in addition to editing the current concept/tag , such as tracing the changes made by users as well as their contributions. A list of others contributors (authors who contributed to the concept) is displayed, to allow for communication between authors.

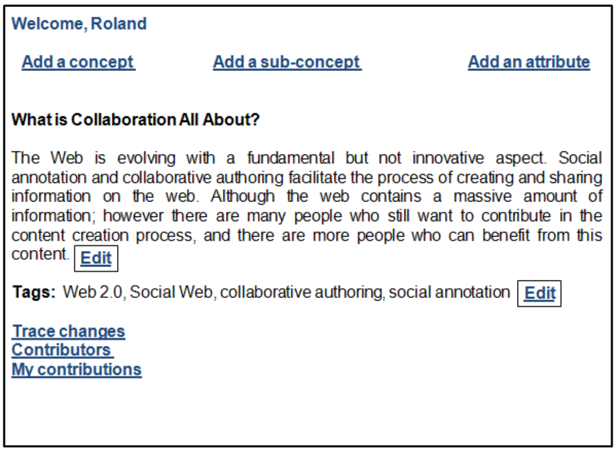

Fig 3. Collaborative authoring in MOT

The student screen is shown in Figure 4: When a student logs into MOS (My Online Student), she can tag the concept, or evaluate it by a rating shown below the concept. Moreover, she can comment on the course, which allows her to ask/answer questions. The user has the option to see all her contributions in addition to all messages sent by her.

Fig 4. Student screen in MOS

6 Related Work

A variety of researches have been done on social annotation in multiple areas such as: folksonomy [11] which we used to define tags as a part of our research; visualization

[7], which we used to visualize the collaboration design; web search [6], which we plan to use as a feature of Collaboration MOT based on the tags; adaptation [1], where Collaboration MOT will “adapt” adaptive materials based on social annotation.

On the other hand, Ahn et al [1] used social annotation to enhance information visualization by defining visual pointers that grant information about a users’ (and a groups’) annotations to web resources; whereas in our work, the system will make recommendations on related materials based on social annotation. For example, if a user annotates a concept then the system will record this annotation and suggest related materials to the user based on this annotation; another example is to display recommendations based on the user’s group interactions, if the user’s group annotates (or contributes in the collaborative authoring) of a concept, then the system will store these collaborations and suggested related materials for all users who belong to that group. Moreover, Bateman et al [2] proposed a structure for combining social annotations (tagging) with natural language ontologies. We argue however, that the tagging system should be based on freely chosen tags rather than applying a pre-defined ontology. However we argue that the best use of co-occurrence values will be covered by evaluating the concepts as described in the new DTD presented in section 5. Finally, Marshall & Brush [9] studied the link between personal and shared annotations, defining the user and group aspects of a system. Therefore, based on such studies, Collaboration MOT will be enriched by adaptation materials based on collaboration between the users.

6 Conclusion and Future Work

In this paper we propose a collaborative design process for authoring of adaptive hypermedia by applying our solutions to the adaptation authoring tool, MOT. We distinguish between two components of collaboration: collaborative authoring (which modifies the actual web resource, i.e., concept in the domain model by multiple authors) and social annotation (which lets the users - adaptive hypermedia authors as well as students - add/edit information without modifying the actual resource). Our basic hypotheses for designing collaboration in adaptive hypermedia, at the levels of authoring and delivery, are confirmed via statistical analysis on a first set of questionnaires with potential users.

Of course, the real challenge is to perform similar studies on the proposed implementation. Users may report wanting one thing in a survey, but behave differently in the actual system. For example, one could expect that unless the ratings are anonymous, or the authors entirely separate from the consumers, the consumers would not want to place honest comments or ratings. It is also not always clear whether students learning material are in the best position to evaluate it; they may mislabel content. Or they may not recognize the utility of the material.

As a next step, the social annotations generated can be exploited by adaptive hypermedia, by reusing this information as possible recommendations for authors and students alike. Therefore, we want to explore the various ways these new annotations can be applied in the adaptation process. Simple pseudo-rules of this kind would be, an adaptation rule that shows students concepts related to the current concept, e.g,:

```
IF (user_accesses_concept)
          THEN (show_other_concepts_with_similar_tags)
```

In this way, the newly contributed, inexpensive content and annotations can be utilized to generate new forms of adaptation and reasoning, thus enriching both authoring and learning experience.

Acknowledgments. The work accomplished in this paper is supported by the EU *ALS* Minerva STREP, the EU FP7 *GRAPPLE* STREP, and was based on the *PROLEARN* Network of Excellence.

References

1. Ahn J., Farzan R., Brusilovsky P. A Two-Level Adaptive Visualization for Information Access to Open-Corpus Educational Resources, Workshop on Social Navigation and Community-Based Adaptation Technologies (AH 06). Dublin, Ireland (2006)
2. Bateman S, Brooks C., McCalla G. Collaborative Tagging Approaches for Onto-logical Metadata in Adaptive E-Learning Systems, Workshop on Applications of Semantic Web Technologies for E-Learning (AH 06). Dublin, Ireland. (2006)
3. Cristea, A. and De Mooij, A. Adaptive Course Authoring: My Online Teacher, ICT'03, Papeete, French Polynesia, IEEE, IEE, ISBN: 0-7803-7662-5, pp. 1762-1769 (2003)
4. Cristea, A. and De Mooij, A. LAOS: Layered WWW AHS Authoring Model and their corresponding Algebraic Operators, WWW'03, The Twelfth International World Wide Web Conference, Budapest, Hungary. (2003)
5. Cristea, A., Smits, D., and De Bra, P. 2007. Towards a generic adaptive hypermedia platform: a conversion case study. Journal of Digital Information (JoDI), Special Issue on Personalisation of Computing & Services, Vol 8, No 3. (2007)
6. Dmitriev, P. A., Eiron, N., Fontoura, M., and Shekita, E. Using annotations in enterprise search. In Proceedings of the 15th International Conference on World Wide Web (Edinburgh, Scotland, May 23 - 26, 2006). WWW '06. ACM Press, New York, NY, 811-817. DOI= http://doi.acm.org/10.1145/1135777.1135900 (2006)
7. Dubinko, M., Kumar, R., Magnani, J., Novak, J., Raghavan, P., and Tomkins, A. 2006 Visualizing tags over time. In Proceedings of the 15th International Conference on World Wide Web (Edinburgh, Scotland, May 23 - 26, 2006). WWW '06. ACM Press, New York, NY, 193-202. DOI= http://doi.acm.org/10.1145/1135777.1135810 (2006)
8. Marshall, C. Toward an ecology of hypertext annotation in Proceedings of ACM Hypertext '98, Pittsburgh, PA. pp. 40-49. (1998)
9. Marshall, C., Brush, A. J. Exploring the Relationship between Personal and Public Annotations. In Proceedings of the ACM/IEEE Joint Conference on Digital Libraries (JCDL04), Tucson, Arizona, June 7-11, 2004, pp.349-357. (2004)
10. Wagstaff, J. A Directory of Social Annotation Tools: http://www.loosewireblog.com/2006/03/a_directory_of_.html, visited on 10/08/2008
11. Wu, X., Zhang, L., and Yu, Y. Exploring social annotations for the semantic web. In Proceedings of the 15th International Conference on World Wide Web (Edinburgh, Scotland, May 23 - 26, 2006). WWW '06. ACM Press, New York, NY, 417-426. DOI= http://doi.acm.org/10.1145/1135777.1135839 (2006)

ANNEX: Questionnaire and Responses

1. Collaborative Authoring in MOT should be designed in such a way that it uses (standards, where applicable) and technologies from:
    - a. 14 Responses: Social Web/ Web 2.0.
    - b. 1 Response: Semantic Web.
    - **c. 15 Responses: Both.**
    - d. 0 Responses: None of the above (it should be proprietary systems only).
2. Social Authoring facilities in MOT is useful if we use annotations in the form of:
    - a. 0 Responses: Rating.
    - b. 1 Response: Feedback.
    - c. 3 Responses: Tags (keywords).
    - d. 0 Responses: Editing content.
    - **e. 26 Responses: All of the above.**
3. Collaborative Authoring facilities in MOT via Social annotation should be done by:
    - a. 2 Responses: Feedback at the level of the concept.
    - b. 3 Responses: Rating at the level of the concept.
    - c. 4 Responses: Tagging at the level of content of attributes.
    - **d. 20 Responses: All of the above.**
4. Social annotation: Feedback, rating & tagging should be applied by
    - a. 5 Responses: Student
    - b. 1 Response: Authors
    - **c. 24 Responses: Both.**
5. Collaborative Authoring in MOT is important for:
    - a. 0 Responses: Collaboration between authors.
    - b. 0 Responses: Collaboration between students.
    - c. 5 Responses: Collaboration between students and authors.
    - **d. 24 Responses: All of the above.**
6. For collaborative authoring, interaction is needed at the level of:
    - a. 4 Responses: users.
    - b. 4 Responses: groups.
    - **c. 21 Responses: Both.**
7. User privileges should be defined at the level of:
    - a. 1 Response: attribute.
    - b. 5 Responses: concept.
    - c. 5 Responses: domain model.
    - d. 7 Responses: lesson.
    - e. 2 Responses: link.
    - **f. 10 Responses: lesson and link.**
    - g. 0 All of the above.
8. Group privileges should be defined at the level of:
    - a. 1 Response: attribute.
    - b. 6 Responses: concept.
    - c. 4 Responses: domain model.
    - d. 7 Responses: lesson.
    - e. 2 Responses: link.
    - **f. 10 Responses: lesson and link.**
    - g. 0 All of the above.

Collaborative Authoring of Learning Elements for Adaptive Learning Spaces

Eric Ras, Jörg Rech, Sebastian Weber

Fraunhofer IESE
Fraunhofer-Platz 1
67663 Kaiserslautern
Germany

eric.ras@iese.fraunhofer.de
joerg.rech@iese.fraunhofer.de
sebastian.weber@iese.fraunhofer.de

Abstract: The *Software Organization Platform* (SOP) intends to support specific software engineering activities such as experience management, requirements engineering, or project management. Wiki pages can be easily used by a transformation engine to produce so-called learning elements. Learning elements are the building blocks of adaptive *learning spaces*, which enhance experience application and understanding in software engineering. This paper shows how learning content is collaboratively authored in the MediaWiki-based SOP in order to generate adaptive learning spaces. The authoring tool, which is embedded as an extension in SOP itself, helps the authors to annotate the learning elements with keywords from the SWEBOK ontology (available in OWL). In addition, a vocabulary manager supports the development of a pre-defined metadata set for annotating learning elements. The authoring environment is a promising technology for solving the problem of "closed content corpus" and uses the advantages of ontologies and semantic relationships in Wikis.

1 Introduction

Web 2.0 concepts (e.g., collaboration, sharing), features (e.g., tagging, folksonomies), and tools (e.g., Wikis, Blogs) support quick and easy sharing of knowledge as well as the creation of learning content in a software organization. Web 2.0 is not only a special technology, but also an umbrella term referring to a class of Web-based applications that make the most of the intrinsic advantages of the Web as a platform. They get better as more people use them by capturing network effects; they harness collective intelligence through user-generated content; they enable collaborative work, and they deliver rich user experiences via desktop-like interfaces [1, 2]. The Software Organization Platform (SOP) connects collaborative generation (i.e., quick and easy page creation and linkage) and semantic annotations (e.g., tagging) of content via a Wiki. In the spirit of Web 2.0, individuals become not only information consumers but also producers. The development of semantic technologies (e.g., Semantic Web techniques, such as OWL, SPARQL, and Web 2.0 techniques, such as tagging, folksonomies, and microformats) offers several possibilities for semantically annotating information and relating chunks of information. Hence, semantic technologies will dramatically change the future development of Adaptive Hypermedia Systems (AHS) and Adaptive Educational Hypermedia Systems (AEHS) in particular. However, AEHS have a common problem that limits the reusability of their adaptive functionality and content. This limitation is due to the design of these systems: The learning resource is usually intertwined with the logic for generating adaptive learning experiences. In addition, adaptive hypermedia systems have worked on a closed set of documents [3, 4]; The documents are fixed at the design stage of the system, and alternations or modifications to the adaptivity are difficult. This closed corpus problem explains why it is difficult to work in an open environment like the Web and profit from the innovations made in the Web 2.0 era. Therefore, new authoring approaches should fulfill the following requirements: separation of adaptivity and content; release of authors (i.e., the software developer or the knowledge engineer) from adaptivity modeling, and easy and flexible annotation of learning content.

To address these problems, an AEHS has been developed to produce so-called context-aware *learning spaces* for enhancing experience reuse and knowledge acquisition in software engineering [5]. The aim of this paper is, first, to show how collaboratively created Wiki content can be transformed into learning content and, second, how it can be annotated by using keywords from an OWL ontology.

2 Vocabulary Management and Learning Content Authoring

Fig. 1 illustrates the process of learning content authoring and learning space generation in SOP. SOP is based on the Semantic MediaWiki [6] and intends to support specific software engineering activities, such as experience management, requirements engineering, or project management. From a technical point of view, a learning space consists of a hypermedia space with linked pages. A learning space follows a specific global learning goal and is created based on context information about the current situation and the context description of an experience package. The learning space is presented technically by means of linked Wiki pages within SOP (step 4). In the following, the first three steps related to authoring will be detailed. A detailed description of how decision models adapt the learning space to the current context (steps 4 and 5 in Fig. 1) can be found in a main conference paper [5]. SOP has shown its usefulness for easy content creation in the past; information about products, processes, roles, groups, customers, organizations, and tools is collaboratively described in the Wiki (step 1). In order to classify these core Wiki pages, Wiki categories (syntax: `[[Category:`*`categoryName`*`]]`) are used to classify the Wiki pages into multiple freely named categories. In addition, by using the features of the Semantic MediaWiki, specific semantic relationships (syntax: `[[`*`relationshipName`*`::`*`wikiPageName`*`]]`) can be defined between instances of the Wiki pages and categories. These relationships are required in the adaptation process to produce context-aware learning spaces. The Wiki pages are stored in the MediaWiki base. Besides these Wiki pages, other kinds of information can be extracted from the Wiki for learning purposes, e.g., definitions, explanations, conclusions, etc. SOP offers an extension to transform these Wiki pages into so-called learning elements, which are the building blocks of learning spaces (step 2). The requirement of easy annotation of learning elements is fulfilled by a set of pre-defined values and metadata attributes for classifying learning elements being offered. This metadata set is defined in SOP by using the *Vocabulary Manager* (see Fig. 2.).

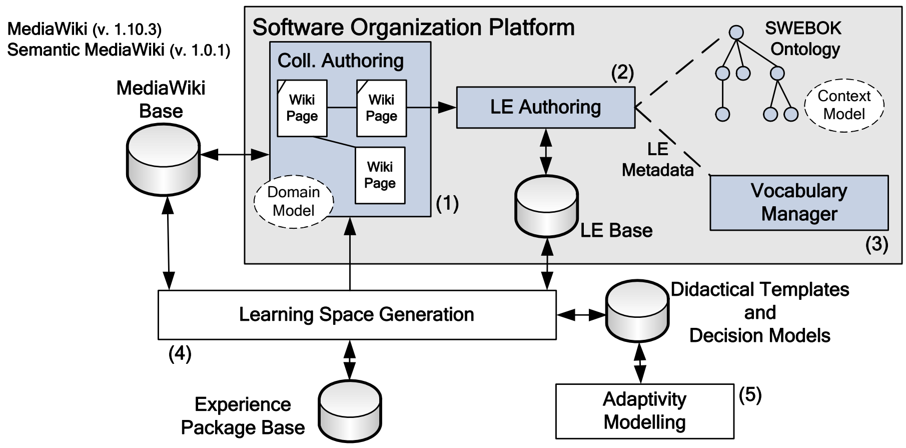

Fig. 1. Software Organization Platform

This SOP extension (step 3) allows creating, editing, and deleting metadata attributes as well as related values (e.g., attribute: illustration; values: example, counter-example).

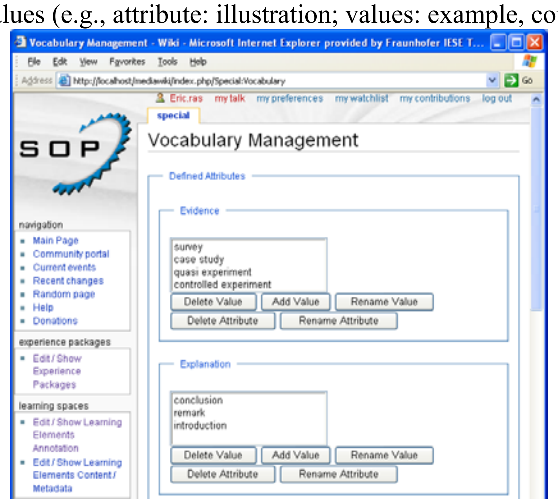

Fig. 2. Vocabulary Manager

In addition to the classification of learning elements, keywords can be used to annotate the learning elements. These keywords are retrieved from a software engineering domain ontology (i.e., an extended ontology based on SWEBOK [7]), from which the semantic relations are also used for the generation of the learning space [5].

The *competence manager* determines the structure (i.e., schema) and the content of the MediaWiki base and is responsible for conceptually developing the context model and the domain model. The *knowledge engineer* has a lot of domain knowledge in terms of packaging and analyzing content. His main task is to extract and annotate content for learning spaces and to instantiate the domain model. The role of the *adaptive instructional design modeler* is to develop instructional design models for selected learning scenarios and to specify variants of the learning space, i.e., to develop the variability model [5]. Hence, the instructional designer must have a strong pedagogical background, knowledge about how to model variants by means of decision models, and knowledge about adaptation methods and techniques.

3 Conclusion

In this approach, the separation of adaptivity modeling and content authoring ensures that both adaptive functionality and content can be reused independently. The software engineer does not have to bother about modeling the pedagogical structures and the adaptivity itself – this is done by the adaptive instructional design modeler. The OWL ontology, which reflects the software engineering concepts and relationships, helps the knowledge engineer to annotate learning elements on the one hand, and supports the generation of the learning space on the other hand. In the future, evaluations need to be done to investigate the usage and acceptance of the authoring tools. Results of the learning space approach can be found in [5, 8].

References

1. Greaves, M.: Semantic Web 2.0. IEEE Intelligent Systems 22, 94--96 (2007)
2. O'Reilly, T.: Web 2.0: Compact Definition? Retrieved May 29, 2008, from http://radar.oreilly.com/archives/2005/10/web-20-compact-definition.html (2005)
3. Brusilovsky, P.: Adaptive Hypermedia. User Modeling and User-Adapted Interaction 11, 87--110 (2001)
4. Henze, N., Nejdl, W.: Extendible adaptive hypermedia courseware: Integrating different courses and web material. In: 1st Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH 2000), Trento, Italy (2000)
5. Ras, E., Ilin, D.: Using decision models for the adaptive generation of learning spaces. In: 5th Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH2008), Hannover, Germany (2008)
6. Semantic MediaWiki. Retrieved May 29, 2008, from http://www.mediawiki.org/
7. Guide to the Software Engineering Body of Knowledge, SWEBOK. IEEE CS Press (2004)
8. Ras, E.: Improving application and understanding of experience packages through learning spaces: A controlled experiment. In: 2nd International Symposium on Empirical Software Engineering and Measurement, Kaiserslautern, Germany (2008)

Modeling Behavior of Users in Adaptive and Semantic-enhanced Information Systems: The role of a User Ontology

Liana Razmerita

Centre of Applied Information and Communication Technologies
Copenhagen Business School, Denmark
lr.caict@cbs.dk

Abstract. The Ontology-based modeling has become a topical subject in the last few years as ontology-based representations can result in better methodologies for conceptual design of data and knowledge bases, facilitating knowledge sharing and reuse. The focus of this paper is ontology-based user modeling and its concrete use in the context of Knowledge Management Systems (KMS). In particular, this paper presents the process of building a user ontology, its integration and use in an ontology-based user modeling framework. This paper proposes a model of user behaviour and it discusses its use for KMS. The paper summarizes our reflections on the role of ontology-based representations for achieving adaptive, personalized features within semantic-enhanced information systems.

Keywords: user modeling, user profiles, personalization, information systems, adaptation, knowledge management, semantic web, ontology.

1 Introduction

In the last few years, the concept of ontology has started to be used frequently in connection with Semantic Web research. An ontology enables the conceptualization and the domain knowledge specification of an application. Ontology aims to structure and represent domain knowledge in a generic way which may be reused and shared across various applications and groups of people. Annotating resources, representing concepts and the relationships between concepts is key for implementing semantic-enabled applications and achieving the Semantic Web vision [1].

Creating flexible, powerful representation knowledge structures on the web is the grounding for achieving advanced, web-enabled, personalized systems. These knowledge structures need to better capture and describe the semantics of data. Ontology-based representations are flexible and powerful representation structures, and they have become a topic of much discussion recently. The process of building an ontology is often a complex, challenging task. It is the first step to achieve semantic web-enabled systems. The complexity lies in its cross-disciplinarity including new techniques, methods and tools. Building a user ontology is a highly interdisciplinary and complex process that requires the expertise in several areas including: knowledge engineering or ontology engineering, software engineering, object-oriented programming, user modeling, artificial intelligence and other domains.

This paper presents the process of building a user ontology as well as its integration and use in an ontology-based user modeling framework. This paper summarizes our reflections on the role of ontology-based representations for adaptive, enhanced-user support. The paper is structured in five sections. The following section describes the process of building the user ontology. The third part describes an Ontology-based User Modeling framework (OntobUMf) prototype built upon the user ontology and it exemplifies how the user profiling can be applied in the context of a Knowledge Management System. The last section concludes and pinpoints towards future work.

2 Ontology-based user modeling

User modelling processes are key for achieving personalized interaction. The personalization process requires access to the user's data and it entails representing, accessing and storing users' related information. The user ontology has been developed based on a top-down approach starting from IMS LIP specification, employing Ushold and Gruninger methodology [2]. The process of building an ontology is divided into three basic steps: capturing, coding, and integrating with existing ontologies. The user ontology has been specified, taking into consideration end-user requirements provided by two Spanish companies involved in development of Knowledge Management Systems (KMS) combined with research of work on user modeling, adaptive hypermedia and user-adaptive interaction and knowledge management. Knowledge Management Systems are information systems dedicated to manage organizational knowledge[3].

At the time the first version of the ontology has been specified, an extensive survey of the user modeling and student modeling literature and its application domains has been done between December and April 2002. According to our findings there was no direct research in the area of user modeling applied in the field of Knowledge Management Systems.

The user ontology is structured according to IMS LIP specification: "*The intent of the specification is to define a set of packages that can be used to import data into or extract data from an IMS compliant learner information server*." [IMS LIP]. IMS LIP package is structured in eleven groupings in order to enable learners to customize their experience and formulate it in a general form. These groupings include the following assimilated concepts: Identification, Goal, QCL (Qualifications, Certifications and Licenses), Accessibility, Activity, Competency, Interest, Affiliation, Security Key and Relationship. According to the IMS LIP specifications, the learner information can be packaged from a variety of systems that are not limited to just Human Resource, Student Information and Learning Management systems.

The concept **Identification** contains attributes and sub-concepts that enable the identification of an individual (name, address, email, etc) within the system. **Affiliation** includes information on the descriptions of the organizations associated with the user/learner. **QCL** contains concepts related to the user's different qualifications, certifications and licenses the user has. **Competency** contains **s**kills associated with formal or informal training or work history. **Activity** includes activities related to the education/training work of the user. **Accessibility** contains concepts related to: user preferences, language information, disabilities etc. The concept **Interest** contains information on hobbies and other recreational activities. The concept **Goal** contains learner's or user's goals, sub-goals and aspirations.

As the top level ontology provided by IMS LIP does not cover the whole features of the user model, we had to extend it with a new concept that model the behavior of the user. **Behavior** is defined as a concept that models characteristics of a user interacting with a system. Behavior concept doesn't exist in IMS LIP package. It is defined as an extension of the existing concepts. Inferred fields grouped as Behavior are calculated based on the data extracted from the log files. For a KMS heuristics and fuzzy logic rules enable to "measure" the **Type_of_Activity**, the **Level_of_Activity** and the **Level_of_KnowledgeSharing** of the users in a KMS. The Type of Activity captures what types of activities a user mainly does; is the user mainly a reader? a writer? or a lurker? Based on their level of activity users are classified as: very active, active, passive or inactive.

- A **reader** is defined as somebody who mainly reads/access the knowledge assets of the system.
- A **writer** is defined as somebody who reads/access the knowledge assets but also submits knowledge assets in the system.
- A **lurker** is defined as somebody who does not contribute and who reads/ access very few knowledge assets in the system.

According to the **level of activity** users are classified as: very active, active, visitor or inactive.

- A **very active** user is somebody who reads, accesses and contributes with knowledge assets.
- An **active** user has less activity in the system then a very active user.
- A **visitor** is somebody who rarely uses the system.
- A person with no activity in the system classified as **inactive.**

According to the level of knowledge sharing the users are classified as: **Unaware, Aware, Interested, Trial,** and **Adopters** inspired by Roger's theory [2] related to diffusion of innovation extended for modeling user's behavior towards adoption of knowledge sharing practices [4].

As a next step in order to realize a computable ontology involves coding it in a formal language. The user ontology has been implemented using ontology editors such as: OntoMat, later using OI-Modeler, KAON and more recently using Protege. KAON is a tool suite for ontology management and for the development of ontology-based applications [5]. It comprises a set of tools and APIs. KAON language is an ontology representation language built on top of RDF/RDFS.

Guarino [6] emphasizes some characteristics of upper level ontologies: they are largely independent of particular applications, they may be possibly language independent, at least within a common culture, they are easily understandable by everybody, in order to be extensively reusable. The next step describes the ontology's integration and use within an ontology-based user modeling framework.

3. Ontology-based user modeling framework

The user modeling techniques and the personalization mechanisms are represented as intelligent services. The architecture of OntobUMf is modular, designed as a service oriented architecture dedicated to user modeling and personalization. The user model data for a specific user is acquired based on an explicit definition, provided by the user, through the user profile editor, and by an implicit part maintained by the category extractor [7-9] represented as intelligent service. Category extractor classifies users based on their activity in the system. As represented below, the activity of the users within the system is captured in the log instances. As depicted in Figure 1, the architecture of OntobUMf integrates the following components:

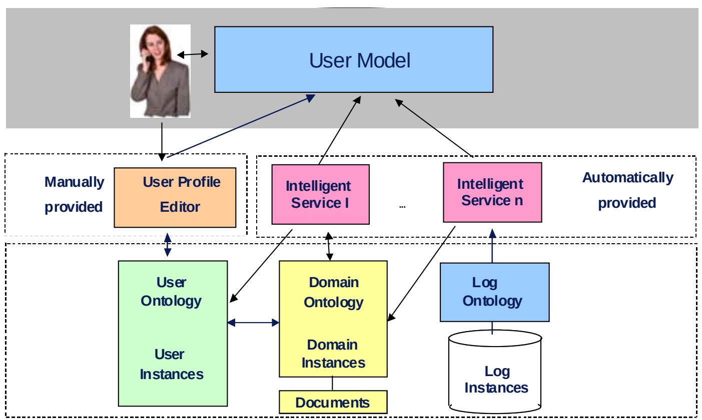

Figure 1 Ontology-based User Modeling framework (OntobUMf) architecture

The **User profile editor** is a specialized ontology editor, dedicated to the end-users to instantiate the user ontology. The user profile editor instantiates the user model but it also enables the user to visualize it, to revise it and update it afterwards. OntobUMf has a modular architecture which enables to add incrementally different intelligent services. Intelligent services can deliver various adaptation methods and personalization techniques. OntobUMf i**ntelligent services** have two main roles in the system:

- to update and maintain the user model on the basis of data available from the running system through the **category extractor.** Category extractor integrates specific mechanisms for modeling the characteristics of the users interacting with a KMS. OntobUMf classify users according to their level of activity, type of activity and level of knowledge sharing.
- to provide **personalized services** based on the characteristics of the users.

OntobUMf is a generic framework however in the following we will outline characteristics of its use in the context of a KMS.

Adaptation methods and personalization techniques relate to specific objectives of KMS. These specific objectives include the following: how to motivate people to create knowledge and submit new knowledge assets in the system, how to stimulate collaboration and knowledge sharing between knowledge workers irrespective of their location, how to alleviate information overload, how to simplify business processes and work tasks.

We define **personalization of a KMS** as the process that enables interface customization, adaptations of the functionality, structure, content and modality in order to increase its relevance for its individual users [10].

The adaptation techniques, at the level of the user interface, can be classified into three categories: adaptation of structure, adaptation of content, adaptation of modality and presentation following Kobsa's taxonomy of personalization. For instance, in the range of adaptation of structure, the system can offer personalized views of corporate knowledge based on interest areas and the knowledge of the users, or its competencies. "Personalized views are a way to organize an electronic workplace for the users who need access to a reasonably small part of a hyperspace for their everyday work." [11]

Adaptation of content refers to the process of dynamically tailoring the information that is presented to the different users according to their specific profiles (needs, interests, level of expertise, etc.). The adaptation of content facilitates the process of filtering and retrieval of relevant information. In a KMS, recommender systems, information filtering agents, and collaborative filtering techniques can be applied with the purpose of adaptation of content. The adaptation of presentation empowers the users to choose between different presentations styles, such as different layouts, skins, or fonts. Other preferences can include the presence or absence of anthropomorphic interface agents, the preferred languages, and so forth. Different types of sorting, bookmarks, and shortcuts can also be included in a high functional system. Adaptation of presentation overlaps in a certain extent with interface customisation. The adaptation of modality enables changes from text to other types of media in order to present the information to the user (text, video, animations, or audio) if they are available in the system.

One of the main objectives of KMS is to make available the knowledge assets: "at the right time to the right people." From this perspective the main sources of personalization are: the user's interests' domain, his/her current goal, his/her work tasks, his/her competencies, etc. We have particularly looked at modeling characteristics of the users' specific to a KMS. KMS need to encourage people to codify their experience, to share their knowledge and to develop an "active" attitude towards using the system. Based on the users' activity in the system, CE infers the user's behavior and it updates certain characteristics of the users interacting with a KMS. A detailed description of the inferences and rules used for modeling the user's behavior can be found in [8, 9]. This behavioral model can be associated with agent-based intervention for the adoption of knowledge sharing behaviors and change management as described in [12].

4 Related Work

The use of ontology for user modelling has been recently proposed for different scenarios: In a ubiquitous computing scenario users can delegate tasks to different agents acting on various devices with computational capability. Context features and situational statements for ubiquitous computing have been proposed as a General User Model Ontology (GUMO) by [13, 14]. The use of user modelling, rules and ontology-based representations for real-time ubiquitous applications in an interactive museum scenario has been proposed by [15]. Dolog and Nejdl, also emphasize the use of ontology for adapted learning content and smart learning spaces. Kay [16] pinpoints to the challenge and need of being able to construct domain ontology automatically and cheaply.

5 Conclusions and Future Work

This paper has presented the process of building a user ontology and its integration and use in an ontology-based user modeling framework. Building a user ontology can be a complex, confusing task for a non-expert in the field.

At the time the first version of OntobUMf ontology was built, we tried to integrate an academic, research-oriented perspective with a more business-oriented perspective. We noticed that the user profiles provided by the two companies involved in the project were different in terms of the terminology employed even though they were from the same country. We observed that many of the user's profile characteristics were synonyms. Trying **to reconcile a different terminology and different world's views can be difficult and in our case the use of a specification facilitated this process**. When beginning the process of building the user ontology, deciding on what concepts to include and how to name them was not simple. In building the user model the strategy was to identify the key user's model characteristics and to identify their associated functionality for the system, in our case a Knowledge Management Systems (KMS). At the beginning we did not try to be exhaustive in identifying all the possible concepts and sub-concepts of the user model ontology. We questioned how these characteristics can be acquired, how to map them into the IMS LIP groupings and what role they will play within the system. The system can capture the user's characteristics explicitly or implicitly. Some of the user's characteristics should be explicitly captured, filled-in by the users via a user profile editor or other means, while other can be inferred based on the user's interaction with the system. Some of the user's dimensions are static while others are dynamic some features change fast while others can change slowly in time. We distinguish between 'must have' and 'nice to have' advanced features of the system.

The use of specifications can help in building the user ontology but it can be limitative as well. **The use of a specification guarantees an agreed-upon a conceptualization and it implies interoperability with other systems compliant with this specification**. Not all the groupings provided by IMS LIP have been employed as concepts, because it was found that not all these concepts were relevant for our specific application domain.

The user ontology has been integrated in OntobUMf user modeling system. The OntobUMf can capture the user's characteristics explicitly or implicitly. Some of the user's characteristics should be explicitly captured, filled-in by the users using a user profile editor while others can be inferred based on the user's interaction with the system. Some of the user's dimensions are static while others are dynamic. Some features change fast (e.g. mood, location) while others can change slowly in time (e.g. type of activity, level of activity, interests, hobby). The Behaviour concept was introduced as an extension of IMS LIP in order to model the behaviour of the user. OntobUMf classify the users according to the level of activity, type of activity and level of knowledge sharing.

Future work involves to extend and test the OntobUMf model of behaviour to classify the users based on their social behavior within communities of practice and social networks. Moreover we plan to define and provide a new set of adaptive, personalized services. Furthermore we plan to apply the same approach of building ontology in an e-government scenario which will be applied within ITAIDE[1] (Information Technology for Adoption and Intelligent Design for e-Government) project.

References

[1] T. Berners-Lee, Hall, W., Hendler, J. A., O'Hara, K., Shadbolt, N. and Weitzner, D. J., "A Framework for Web Science. Foundations and Trends in Web Science," vol. 1, pp. 1-130, 2006.

[2] M. Uschold and M. Gruninger, "Ontologies: principles, methods, and applications," *Knowledge Engineering Review*, vol. 11, pp. 93-155, 1996.

[3] M. Alavi and A. Tiwana, "Knowledge Management: The Information Technology Dimension," *Blackwell Handbook of Organizational Learning & Knowledge Management*, pp. 104-121, 2003.

[4] A. Angehrn, Nabeth, T., Razmerita, L., Roda, C., , "K-InCA: Using Artificial Agents for Helping People to Learn New Behaviours," vol. Proc. of IEEE International Conference on Advanced Learning Technologies (ICALT 2001), 2001.

[5] A. Maedche, B. Motik, L. Stojanovic, R. Studer, and R. Volz, "Ontologies for Enterprise Knowledge Management " *IEEE Intelligent Systems*, vol. November/December, 2002.

[6] N. Guarino, "Formal Ontology and Information Systems," in *Proc. of the 1st International Conference, In N. Guarino (ed.)*, Trento, Italy, 6-8, 1998.

[7] L. Razmerita, Angehrn A. & Maedche, A., , "Ontology-based User Modeling for Knowledge Management Systems," in *Proceedings of the Ninth*

[1] http://www.itaide.org

International Conference of User Modeling 2003, Pittsburgh, USA, 2003, pp. 213-217.
[8] L. Razmerita, "Ontology-based User Modeling for Knowledge Management Systems," in *Ontologies: A Handbook of Principles, Concepts and Applications in Information Systems*, R. K. a. R. R. Raj Sharman, Ed.: Springer, 2007, pp. 635-664.
[9] L. Razmerita, "User Models and User Modeling in Knowledge Management Systems: An Ontology-based Approach." vol. PhD Toulouse: Paul Sabatier University, France, 2003, p. 247.
[10] L. Razmerita, Angehrn, A., Nabeth, T., Roda, C.,, "InCA: An Intelligent Cognitive Agent-based Framework for Adaptive and Interactive Learning," in *Conference on Cognition and Exploratory Learning in Digital Age, IADIS CELDA 2004*, Lisbon, Portugal 2004.
[11] P. Brusilovsky, "Adaptive educational systems on the world-wide-web: A review of available technologies," in *In Proceedings of the 4th International Conference in Intelligent Tutoring Systems*, San Antonio, Texas, USA, 1998.
[12] C. Roda, Angehrn, A., Nabeth, T., Razmerita, L., "Using conversational agents to support the adoption of knowledge sharing practices," *Interacting with Computers, Elsevier*, vol. 15, pp. 57-89, 2003.
[13] D. Heckmann, Schwartz, T., Brandherm, B., Schmitz, M., and von Wilamowitz-Moellendorff, M., "GUMO - the General User Model Ontology," in *10th International Conference on User Modeling (UM'2005)*, Edinburgh, UK, , 2005, pp. 428-432.
[14] D. Heckmann, Schwarzkopf, E., Mori, J., Dengler, D., and A. Kroner, "The User Model and Context Ontology GUMO revisited for future Web 2.0 Extensions." vol. Contexts and Ontologies: Representation and Reasoning, 2007, pp. 37-46.
[15] M. Hatala, R. Wakkary, and L. Kalantari, "Rules and ontologies in support of real-time ubiquitous application," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 3, pp. 5-22, 2005.
[16] J. Kay and A. Lum, "Ontology-based user modelling for semantic web," in *Workshop 8 associated with the 10th International Conference on User Modeling (UM'05)*, Edinburg, UK, 2005, pp. 11-19.

Adaptive ontology-based navigation

Petr Šaloun[1] and Zdenek Velart[2]

[1] Department of Informatics and Computers, Faculty of Science, University of Ostrava
petr.saloun@osu.cz

[2] Department of Computer Science, Faculty of Electrical Engineering and Computer Science, VSB Technical University Ostrava
zdenek.velart.fei@vsb.cz

Abstract. This paper presents a proposal of navigation for the user, based on an existing ontology of the domain of the learning course. The navigation scheme of the course stands on interconnection of an ontology with existing learning material by the use of the keywords of the domain. On one side there is the user's needs and goals and on the other side are possibilities and limitation of existing systems.

1 Introduction

This paper presents our approach and results of research of navigation in a learning course. Navigation of the student in the information space is a very important task. We narrow our interest on course-based learning where the navigation is learning-oriented. We think that the student should be directly navigated to the desired information, which is seeking or which he wants to learn. According this we base our navigation scheme on an existing ontology of the domain, where relationship between knowledge is stored. The interconnection of the ontology with an existing course is on keywords of the domain.

As an integral part of the navigation a problem of user model initialization must be discussed – a problem of users first visit to the course. In our navigation scheme we discuss several possibilities how to deal with such a situation.

The basics of this idea were presented in [6]. Our problem domain are the courses of Programming in C++ and Electronical publishing. We taught the C++ course for two years (our results are presented in [5]) with about 400 student by the use of the adaptive hypermedia system AHA! [2]. The course is organized into twelve chapters, with 101 concepts. In our case every concept is realized by one XHTML file. The ontology we used we defined according the ISO specification of the C++ programming language and consists of about 300 elements.

2 Ontology-based navigation

The basic idea of the ontology-based navigation stands on the premise: *there is necessary to know all prerequisites to learn new information.* Every concept of the domain (which can be defined and explained in paragraph, page or set of pages) is represents by the learning material of the domain.

The structure of the domain is captured in the domain ontology, where relationships and dependencies between elements are stored. Therefore the use of an ontology for the navigation in the course is preferable. Apart from the primary idea in [6] particular concept in the course is described by three distinct sets – *prerequisite set*, *outcome set* and *inout set*. These sets consists of the elements of the ontology. Element placed in *prerequisite set* represents prerequisites of the concept. Elements in *outcome set* represents what student can learn in the concept. Elements are placed into *inout set* when they should be at the same time in *prerequisite* and *outcome* sets. This indicates that the concept is dealing with "extended level" of the knowledge. Student's knowledge learned from such a concept is on higher level than from a basic one.

2.1 Navigation

The information about visited concepts is stored in the student's personal profile. The system tracks for the user which concept he visited and stores the elements from these concept. When the student learns a new concept all the elements form the *outcome set* are added to his *achieved knowledge set*, which is used in the navigation as is described later in the paper.

Presentation of concepts is based on the student's actual goal, preferences and knowledge stored in his profile. Our model stands on the reference AHAM model, where user's profile together with adaptation rules and engine are used for the actual presentation of the content.

Let us now discuss the navigation for the student in the course. The navigation is described by Figure 1 – used numbers corresponds with the description in the list:

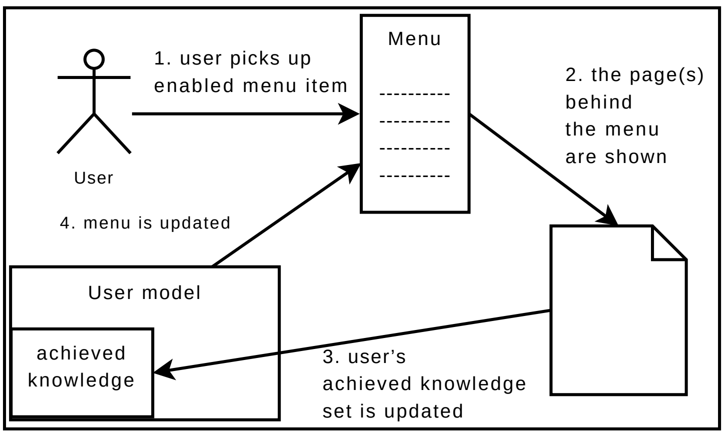

Fig. 1. Course navigation

1. Student chooses a concept from available concepts in the menu, which is presented to him.
2. According the chosen menu item the student is presented with learning material of the particular concept.
3. When student finishes the learning material, the system can test the students knowledge of the concept – this step is optional. The system then updates student's *achieved knowledge* set with the elements from the *outcome* set of the concept.
4. Student's menu is updated according his actual *achieved knowledge* set. Concepts which all elements from the *prerequisite set* are already in the achieved knowledge set are enabled. Navigation continues with the step 1.

The concepts presented to the student are ordered – on the first place, there are links to concepts for which the student has already all the necessary prerequisites or if achieves defined threshold value. Optionally other links can be presented – links to already visited concepts and not clickable links (only names) to concepts for which the student does not have all the necessary prerequisites.

For better navigation there is also possibility to construct a visual "map" of the course. In this map the current path of the student is highlighted. Concepts for which the students has all the necessary prerequisites are also rendered as clickable.

3 User model initialization

User model initialization represents a situation where the student visits the course for the first time. In such a moment the system have to set initial values for the newcomer.

- **Basic settings** – the student is presented only with concepts with empty prerequisite set or directly defined starting concepts of the course.
- **Knowledge test settings** – the student takes a "knowledge test" where the system tests the actual users knowledge about the concept.
- **Domain settings** – if the student comes from a similar course from the same domain (for example a student wants to take a C++ course afther finishing Java course) the system can reuse what the student have already learned. Mapping between the two ontologies is necessary in this case.
- **Course subset settings** – if the students comes to the system only with limited amount of goals, needs only to learn a limited amount of the information, the system can offer to the student only a subset of the concepts of the course, which lead to the desired goal. The system needs to make a subset of the concepts, starting from the goal and to choose concepts where the element is in outcome set. For the elements in the prerequisite set the system must find concepts, where the elements are in outcome set. This step is needed to do recursively for all found concepts.

4 Related works and conclusions

In this paper we presented our proposal of an ontology-based course navigation. Similar problem as ours can be found in other papers. One of these is [3], where authors present a visual education tool for efficient and effective learning. The prerequisite dependence relationship between concepts is based on extracted concept definitions. The authors assume existence of the concept definition database of the domain for the construction of the concept graph, which is then used for the navigation. Occurrences of one concept name in the others concept definition is used as a base for interconnection of the concepts. On the contrary our approach is based on the use of existing ontology of the domain. The difference between these two approaches is in what is used as the source for creating the relationships between concepts. Even if our approach is based on the ontology (in case of our C++ course build from ISO specification) the approach and course creation is transparent to the author of the course.

The authors in [1] presents a method aimed at creating content represented by an ontology and exporting such a content to a existing adaptive applications. Their approach is based on use of the core ontology, which is defined in the paper and is designed for adaptive application content modeling.

Our future research in this area will focus on extending the definition of the ontology-based course structure with situation where the course is enhanced with additional learning material or new concepts or on the other hand some material or concepts are removed from the course. We are also dealing with the automatization of the course creation process by the use of classification – some hints can be seen in [4]. We also prepared our C++ course with the described navigation together with a web-based system which is able to deliver such a course. The course will be offered to our students. After the run of the course we would like to compare our approach with data gathered in previous years, where the navigation in the course was mainly sequential with predefined fixed order of concepts.

References

1. Bieliková M., Moravčík M. Modeling the Content of Adaptive Web-Based System Using an Ontology. *In proc. of* 1^{st} *International Workshop on Semantic Media Adaptation and Personalization SMAP06.* IEEE Computer Society, pp 115-120.
2. De Bra P., Aerts A., Berden B., De Lange B., Rousseau B., Santic T., Smits D., Stash N. AHA! The Adaptive Hypermedia Architecture. *In Proceedings of the ACM Hypertext Conference*, Nottingham, UK, August 2003, pp. 81-84
3. Godbole S., Joshi S., Mehta S., Ramakrishnan G. Toward interactive learning by concept ordering. *In HT'07 Proceedings.* Manchester, UK, 2007, pp. 149-150.
4. Nepil M., Popelínský L.: Committee-Based Selective Sampling with Parameters Set by Meta-Learning. *In Proc. of ECML/PKDD Workshop on IDDM'02* Helsinki
5. Šaloun P., Velart Z. Adaptive Hypermedia as a mean for learning programming. *In Workshop proceedings of the* 6^{th} *ICWE 2006.* ACM Digital Library, 2006.
6. Velart Z., Šaloun P. Ontology Based Course Navigation. *In HT'07 Proceedings.* Manchester, UK, 2007, pp. 151-152.

Management of learning styles, competences and access device preferences to alleviate the authoring of standard-based adaptive learning designs

Silvia Baldiris[1], Olga C. Santos[2], Germán Moreno[1], David Huerva[1], Ramón Fabregat[1], Jesús G. Boticario[2]

[1] Insitute of Informatics and Aplications (IIiA), Universitat de Girona, Spain
{baldiris, lauram, gmoreno}@eia.udg.edu, ramon.fabregat@udg.edu
http://bcds.udg.edu/
[2] aDeNu Research Group, Artificial Intelligence Department, Computer Science School, UNED, C/Juan del Rosal, 16. 28040 Madrid, Spain
{ocsantos, jgb}@dia.uned.es
http://adenu.ia.uned.es/

Abstract. The complexity of authoring adaptive learning designs and the extra effort and technical knowledge required for course designers restrains the actual application of adaptive course delivery in real settings. Our research works focuses on alleviating the workload for the course designer by applying artificial intelligence techniques that automate part of the design process. We have designed an integrated user model based in an intensive use of IMS learning specifications for the management of learning styles, competences and access device preferences. This model is used by some adaptation processes that apply artificial intelligence techniques to produce an automated learning design, which enters in the life cycle of the learning process, which covers design, publication, use and auditing of learning processes.

1 Introduction

Authoring of adaptive and adaptable hypermedia can benefit from the application of existing standards and specifications (e.g. reusability and interoperability of designs among different systems and extensions of current design with feedback from its delivery). A review of the state of the art in this field was done in [1] and reported that the 1) authoring task has been the major bottleneck for decades and 2) the main problem in designing standard-based adaptive courses is the complexity to establish the hooks for the dynamic modelling to be performed at runtime. In ADAPTAPlan [2] project we have been working to provide dynamic assistance to course authors by relaxing their design work to provide a simplified set of data (course objectives, questionnaires, contents, services and activities) which is modelled with IMS specifications [3]. With these data, the learning design of the course, in terms of the IMS Learning Design specification can automatically be built. In this paper, we present current works considering the device access preferences when building the learning design of the course, not considered previously.

2 Modeling support for the adaptation process

In order to generate adaptations that alleviate the course design task, users are modelled in ADAPTAPlan in terms of competences, learning styles and access device preferences (see. Fig.1). The data are obtained form the learners interactions in the learning platform. In particular, ADAPTAPlan is implemented on an open source learning platform called dotLRN [4], which supports IMS educational standards, provides accessibility features, and is technically designed to be extensible and supports web services communication. Data to generate the *Learning Style Profile* is computed by applying Felder Learning style inventory [5]. Moreover, an external service to manage user model features following IMS Learner Information Profile (IMS-LIP) is being implemented [6]. In this way, the user learning style is stored as the corresponding user properties in the IMS –LIP implementation. This service also manages the access preferences of the user in terms of IMS Accessibility for LIP specification, so these preferences can also be taken into account at design time in order to adapt the learning design of the course to the user.

Access Device Profile can be queried through an external CC/PP UAProf repository [7]. dotLRN platform is being prepared to extract from the HTTP header the information of the device used. External repositories can be used, such as [8]. Moreover, there are also on-going works integrating a device profile server into dotLRN [6].

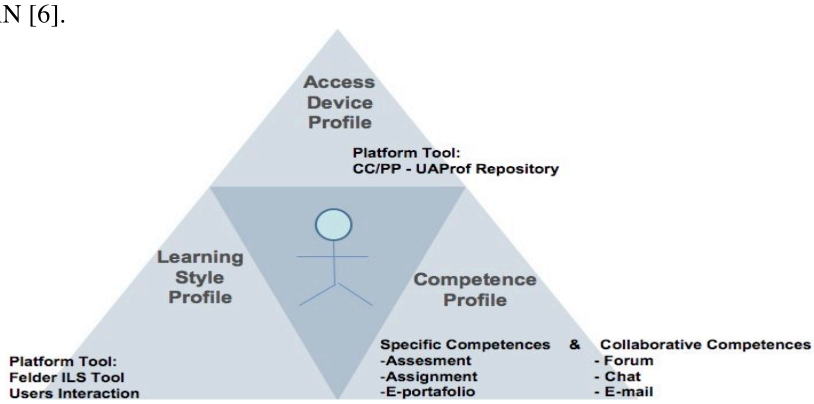

Fig. 1. Interrelations among Learning Styles, Competences and Access Devices

In order to obtain the *Competence Profile*, several educational tools are offered by the platform. In particular, the IMS Question and Test Interoperability is supported by the assessment package which computes the achievements of the user in a standard-based format. Other packages such as assignment and e-portfolio are the base to define specifics competences model; and data from forum package, chat package and e-mails package are useful for the collaborative competences model.

Several implementations can be used for modelling the learning process and in particular, are suitable for supporting the adaptation process required: 1) LORS Package: it is an implementation of Scorm Reference Model [9] and includes an implementation of IMS Metadata and IMS Content Packaging; 2) IMS Reusable Definition of Competency or Educational Objective is used to specify a catalogues of learning goals [10]; 3) IMS Learning Design Player delivers the adapted learning design. The relationships between IMS specifications are the base to generate adaptation rules [2].

3 Adaptations to automate the authoring process

The purpose of the adaptation mechanism in ADAPTAPlan project defines: 1) the best type of learning resources and the order to present them; 2) the collaborative activities that should be presented to user according with his/her collaborative competences level; 3) the learning resources, no collaborative activities and evaluation resources to deliver according to specific level of competences; and finally, 4) changes on the platform interface and the selection of some learning objects according to the CC/PP profile associated to the learner access device. Figure 2 shows the main elements and characteristics of the adaptation process [11].

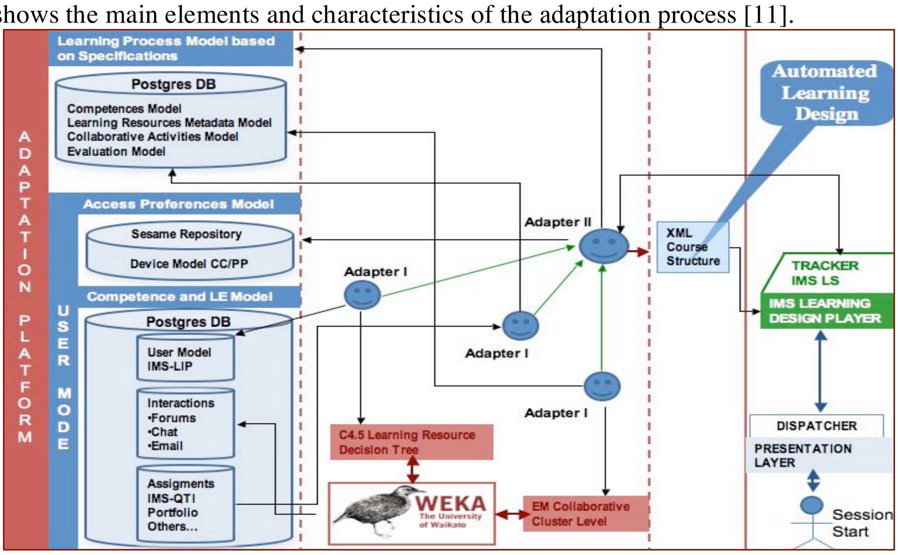

Fig. 2. Adaptation Process ADAPTAPlan Project

With the *Learning Style Model* stored in the IMS Learner Profile as input, Adapter I selects the best order to present learning resource types according to the learning style information. Following previous experiences, this information is stored in a C 4.5 Learning Resources Decision Tree. The *Collaborative model* is developed using database information about the learner's interaction in the collaborative tools. Data is pre-processed and the EM algorithm (Weka Tool) is utilized in order to generate clusters of users according to the collaboration level, as previous experiences in [12]. With the generated EM model, Adapter I process determines in which cluster the student is located and defines the specifics resources to show him.

Adapter II, dynamically and on a second phase, performs the access device adaptation by adjusting learning objects characteristics such as image resolution, weight, colours, dimensions and another measure associated to resources presentation and the html structure, and ignores some inadequate resource for a specific device. This adaptation is based in information available in the metadata object repository. The result of adaptation process is an IMS Learning Design.

6 Conclusions

We have presented some advances achieved within the ADAPTAPlan project, which extend the user features to consider the access device capabilities.

User model data were presented, such as a collection from a pervasive usage of standards and specifications (IMS family of specifications) and from different tools available in a learning platform.

In order to support the learning style and competences adaptation, machine learning techniques are applied. Decision about the order to present learning resources is based on a classification C 4.5 algorithm and the cluster of competences level are composed using EM algorithms.

References

1. Boticario, J.G., Santos, O.C. (2007) A Dynamic assistance approach to support the development and modelling of adaptive learning scenarios based on educational standards. Fifth International Workshop on Authoring of Adaptive and Adaptable Hypermedia. International Conference on User Modelling 2007 (in press).
2. Baldiris, S., Santos, O. C., Barrera, C., Boticario J. G., Velez, J., Fabregat, R.: Integration of educational specifications and standards to support adaptive learning scenarios in ADAPTAPlan. International Journal of Computer Science and Applications (IJCSA). Special Issue on New Trends on AI techniques for Educational Technologies. Vol 5, 1, 2008.
3. IMS GLC specifications: http://www.imsglobal.org/
4. Santos, O.C., Boticario, J.G., Raffenne, E., Pastor, R. Why using dotLRN? UNED use cases. FLOSS International Conference, 2007.
5. Felder R. M., Silverman L. K., 'Learning and Teaching Styles In Engineering Education', Engr. Education, 78(7), 674–681 (1988) – Preface: Felder R. M., June 2002.
6. Cuartero, A., Santos, O.C., Granado, J., Raffenne, E. and Boticario, J.G. Mangement of standard-based User Model and Device Profile in OpenACS. Proceedings of the International Conference and Workshops on Community based environments, 2008.
7. CC/PP Information Page, (http://www.w3.org/Mobile/CCPP/)
8. Mérida, D., Cannataro, M., Fabregat, R. and Arteaga, C.: "MAS-SHAAD a Multiagent System an Adaptive Hypermedia System". Proceedings of IJCEELL journal Special issue: Adaptivity in Web and Mobile Learning Services, (2004).
9. SCORM 2004, Sharable content Object Reference Model, Impact Summary. Versión 1.0. www.ADLNet.gov.
10. Silvia Baldiris, Ramón Fabregat, Olga Santos. "Modelling Competency upon dotLRN". OpenACS and .LRN Spring Conference. Viena. April 2007.
11. Santos, O.C., Baldiris, S., Velez, J., Boticario, J.G., Fabregat, R. Dynamic Support in ADAPTAPlan: ADA+. Proceedings of CAEPIA. (Eds.) Borrajo, D., Castillo, L. and Corchado, J.M. Actas de la XII Conferencia de la Asociación Española para la Inteligencia Artificial. Vol. II. 2007, p. 131-140.
12. Gaudioso, E., Boticario, J.G.: Towards web-based adaptive learning communities. In Artificial Intelligence in Education, ed. by H.U. Hoppe et. al. (IOS Press, Amsterdam). pp. 237-244, 2003.

Early Design Performance Prediction for Authoring Web-based Hypermedia Applications

Sanjeev Dhawan*, Rakesh Kumar**
*Faculty of Computer Engineering, University Institute of Engineering & Technology (U.I.E.T), Kurukshetra University, Kurukshetra (K.U.K)- 136 119, Haryana, India.
**Faculty of Computer Science, Department of Computer Science and Applications (D.C.S.A), Kurukshetra University, Kurukshetra (K.U.K)- 136 119, Haryana, India.
E-mail: {rsdhawan,rsagwal}@rediffmail.com

Abstract

This paper has been designed to identify the Web metrics for evaluating the reliability and maintainability of hypermedia applications. In the age of Information and Communication Technology (ICT), Web and the Internet, have brought significant changes in Information Technology (IT) and their related scenarios. Therefore in this paper an attempt has been made to trace out the Web-based measurements towards the creation of efficient Web centric applications. The dramatic increase in Web site development and their relative usage has led to the need of Web-based metrics. These metrics will accurately assess the efforts in the Web-based applications. Here we promote the simple, but elegant approaches to estimate the efforts needed for designing Web-based applications with the help of User Behavior Model Graph (UBMG), Web page replacement algorithms, and RS Web Application Effort Assessment (RSWAEA) method. In this paper, variations of the effort estimations, for writing code for web-based applications, are explored. Effort assessment of hyperdocuments is crucial for Web-based systems, where outages can result in loss of revenue and dissatisfied customers. Here we advocate a simple, but elegant approach for effort estimation for Web applications from an empirical point of view. The proposed methods and models have been designed after carrying out an empirical study with the students of an advanced university class and Web designers that used various client-server based Web technologies. Our first aim was to compare the relative importance of each Web-based metric and method. Second, we also implemented the quality of the designs obtained based by constructing the User Behavior Model Graphs (UBMGs) to capture the reliability of Web-based applications. Thirdly, we use Web page replacement algorithms for increasing the Web site usability index, maintainability, reliability, and ranking. The results obtained from the above Web-based metrics can help us to analytically identify the effort assessment and failure points in Web-based systems and makes the evaluation of reliability of these systems simple.

Keywords: Web metrics, Web page replacement algorithms, Web usability, RS Web application effort assessment (RSWAEA), User behaviour model graph, method (UBMG).

1. Introduction

Reliable and precise effort estimation of high volume Web-based hyperdocuments is critical for project selection, planning and control. Over the past thirty years, various estimation models have been developed to help the managers to perform estimation tasks, and this has led to a market offering a number of estimation tools. For organizations interested in using such estimation tools, it should be crucial to know about the predictive performance of the estimates such tools produce. The construction of an estimation model usually requires a set of completed projects from which an arithmetic model is derived and which is used subsequently as the basis for the estimation of future projects. So, there is a need for an estimation model to identify efforts of these Web projects. Web hypermedia applications have great potential in areas such as software engineering, education, and training to mention but a few. This paper looks at the relative importance of Web-based metric, their design, reliability, Web site usability and maintainability index, and ranking. The results obtained from the above techniques can easily evaluate the size and other important attributes related to Web-based hypermedia applications. The Web-based hypermedia applications are the non-conventional applications characterized by the authoring of information using nodes (chunks of information), links (relationship between nodes), access structures (for navigation), anchors, and its delivery over the Web. Web technologies commonly used for developing such applications are HTML, JavaScript, PHP and multimedia. The World Wide Web has created a standardized communications infrastructure that has enabled a wide range of applications, especially for business-to-business eCommerce, customer support, and entertainment.

The rapid design and deployment of web applications has been done largely in the absence of performance considerations. In addition, there have been great difficulties with forecasting site access patterns and dealing with the scalability issues. Thus, it is not surprising that web-based applications frequently experience problems with poor reliability, long response times, and other important issues. This paper describes Web-based metrics for detecting and resolving such problems. Consequently, this paper has two main objectives: the first is to design a reliable and cost effective design using UBMG via Web server log files as prime attributes. The second objective is to compare the Web page replacement algorithms for defining the usability and maintainability index to estimate the effort to develop Web-based hypermedia applications, and finally, to choose the one that gives the best results, according to several measures of accuracy. Furthermore, an empirical research and study was carried out to provide effort assessment for small to large-size Web-based applications. For this paper, we have analyzed many findings drawn from the related questionnaire. The results are designed by answers of questionnaire from a survey of the students of an advanced university class and web designers. Our analyses suggest several areas (including reliability, usability, maintainability, complexity, cost, time requirements and type of nature of Web design) where both Web-based designers, engineers and managers would be benefitted from better guidance about the proper implementation of Web-based applications.

2. Qualities of good software metric

Lord Kelvin once said that when you can measure what you are speaking about and express it in numbers, you know something about it. Measurement is fundamental to any engineering discipline. The terms "measure", "measurement", and "metrics" are often used interchangeably, but according to Pressman [1] a measure provides a quantitative indication of the extent, amount, dimensions, capacity, or size of some attribute of a product or process. Measurement is the act of determining a measure. The IEEE Standard Glossary of Software Engineering Terms [2] defines metrics as "a quantitative measure of the degree to which a system, component, or process possesses a given attribute". Ejiogu [3] suggested that a metric should possess the following characteristics: (a) Simple and computable: It should be easy to learn how to derive the metric and its computation should not be effort and time consuming, (b) Empirically and intuitively persuasive: The metric should satisfy the engineer's intuitive notion about the product under consideration. The metric should behave in certain ways, rising falling appropriately under various and conditions, (c) Consistent and Objective: The metric should always yield results that are unambiguous. The third party would be able to derive the same metric value using the same information, (d) Consistent in its use of units and dimensions: It uses only those measures that do not lead to bizarre combinations of units, (e) Programming language independent, (f) An effective mechanism for quality feedback. In addition to the above-mentioned characteristics, Roche [4] suggests that metric should be defined in an unambiguous manner. According to Basil [5] Metrics should be tailored to best accommodate specific products and processes.

3. Motivations of UBMG and RSWAEA

The techniques we propose have the following key objectives: (a) Derive the UBMG in a manner that capture complete details for valid sessions, and number of occurrences of invalid sessions, and (b) Derive the RSWAEA method to estimate the development effort of small to large-size projects, especially in scenarios that require fast estimation with little historical information. The valid sessions have metrics such as session count, reliability of session, probability of occurrence of the session, and transition probability of the pages in the session. On the basis of RSWAEA method, the Web-based software effort estimations are examined with user's cost, cost drivers, data Web objects, compatibility, usability, maintainability, complexity, configuration, time requirements, and number of interfaces.

3.1 Implementation and Analysis of User Behavior Model Graph (UBMG)

UBMG can be represented in form of a graph or a matrix notation [6]. In the graph view, nodes represent the pages, and arcs represent the transition from one node to another. In the matrix representation each cell (*i,j*) corresponds to probability of transition from page *i* to page *j*. We extend UBMG by adding an additional node to the graphical view, and a column in case of the matrix view to represent errors encountered while traversing. The construction of UBMG starts with the navigational model and access logs as described in [7], where the navigational model represents the complete overview of the different pages and the flow between the pages in the Web system. The access logs store information regarding the timestamp, page accessed client-id, referrer-id, HTTP return code etc. for determining session information. A sample format of IIS log file is shown in figure1.

```
#Fields: date time c-ip s-port cs-uri-stem cs-uri-query
sc-status time-taken cs (User-Agent) cs (Referrer)
```

Fig. 1. Format of IIS server log files

```
<Date and Time> <Client-id> <URL> <Referrer-id>
  2008-03-25 00:00:00 201.124.225.77 a.asp
  2008-03-25 00:00:02 201.124.225.77 b.asp
  2008-03-25 00:00:03 201.124.225.77 c.asp
  2008-03-25 00:00:05 201.124.225.77 e.asp f.asp
  2008-03-25 00:00:07 201.124.225.77 f.asp a.asp
  2008-03-25 00:00:06 201.124.225.77 d.asp f.asp
  2008-03-25 00:00:05 201.124.225.77 f.asp g.asp
  2008-03-25 00:00:10 201.124.225.77 d.asp g.asp
  2008-03-25 00:00:06 201.124.225.77 f.asp h.asp
  2008-03-25 00:00:06 201.124.225.77 a.asp h.asp
```

Fig. 2. Access log entries of IIS server

We consider referrer-id and the client-id fields as the basis to do a depth first search on the access logs. This approach will segregate valid and invalid sessions. To understand, consider an application with only two independent sessions- S1 with pages (a→ b→ c→ d → f→ g) and S2 with pages (a→ b→ c → e→ f→ h). Let the access log have entries as shown in figure 2. We derived two valid sessions when depth first search is based on client-id fields. However, with the referrer-id field we determine the invalid path consisting of pages (a → b→ h). The count of all such invalid sessions is determined, and the construction of UBMG is done only for the valid sessions. Let us consider the example of an Online Airline Reservation System (OARS), where the two sessions defined in the navigational model are Session 1 (S1):"Book a seat" with pages SeatSelection.asp→ SeatDetails.asp→SeatBooking.asp→SeatConfirmed.asp→ SeatUnconfirmed.asp→ReservationControlNumber.asp→ Payment.asp. Session 2 (S2): "Cancel a seat" with pages SeatSelection.asp→SeatDetails.asp→SeatBooking.asp→ SeatConfirmed.asp→SeatUnconfirmed.asp→ReservationControlNumber.asp→Refund.asp. We tag an alias for the pages as given in figure 3, 4 and 5.

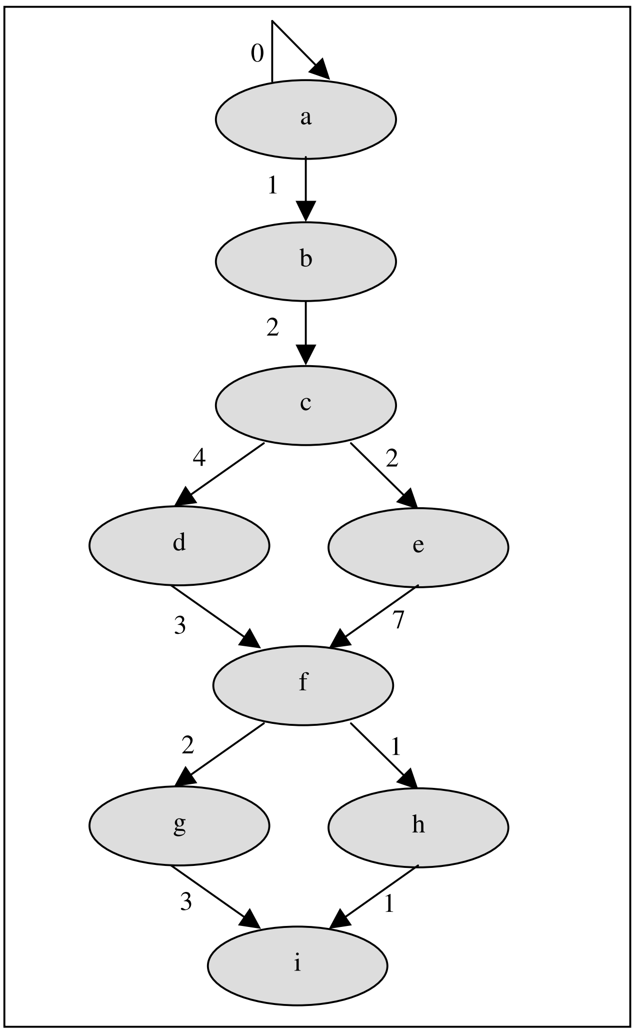

Fig. 3. Graphical view of UBMH

Where: a-SeatSelection.asp; b-SeatDetails.asp; c-SeatBooking.asp; d-SeatConfirmed.asp; e-SeatUnconfirmed.asp; f-ReservationControlNumber.asp; g-Payment.asp; h- Refund.asp. Figure 3 shows the graphical view of UBMG with the exit node 'i'. The matrixes of transition probabilities for graphs 3 and 4 have been shown in table 1 and 2 respectively. The matrix of table 1 considers only those sessions that have completed successfully, and the matrix of table 2 considers both successful sessions and sessions related to error nodes. For example, sum of probabilities of the paths out of the node b is 1 (table 1, 2 and 3) indicating that 10% of clients had either dropped out or encountered errors.

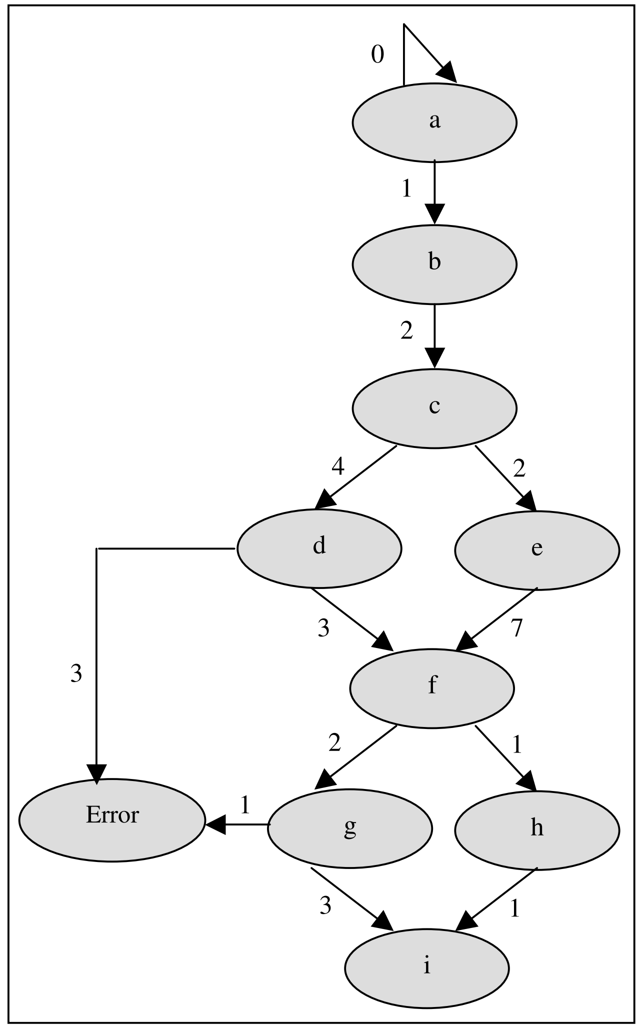

Fig. 4. Addition of an error node to UBMH

The probability of reaching a node *j* in the graph can be calculated using Markov property [7, 8, 9]. The generalized notation of using Markov property is:

$$Nj = N1 * P(1,j) + N2 * P(2,j) + \ldots + Nk * P(k,j) \quad \textbf{(1)}$$

Where, k is the number of nodes that lead to node j. In the OARS example of table 1 (figure 3), to compute the probability of reaching the node 'f' is 3 * Nd + 7 * Ne and probability of reaching the node 'i' is 3 * Ng + 1 * Nh, where Na is equal to one. In the OARS example of table 2 (figure 4), to compute the probability of reaching the error node 'Error' is 3 * Nd + 1 * Ng. So, when an error node is included, then sum of all the incoming and outgoing weights of edges is increased. In the OARS example of table3 (figure 5), to compute the probability of reaching the error node 'Error' is 3 * Na + 3 * Nd + 1 * Ng. Finally, from the table 1, 2 and 3; it has been observed that whenever there is a existence of new node (either error or virtual node) the probability of reaching the new node is increased. Therefore, to resolve such type of issues, we have developed the Web replacement policies/algorithms (as discussed in section 4) to enhance the usability and reliability indexes of Web pages stored at different Web servers. The complexity of figures 3, 4, and 5 can be calculated using cyclomatic complexity (e-n+2 or e-n+1 or numbers of two edges in a single node+1, where: e is the total number of edges and n is the total number of nodes).

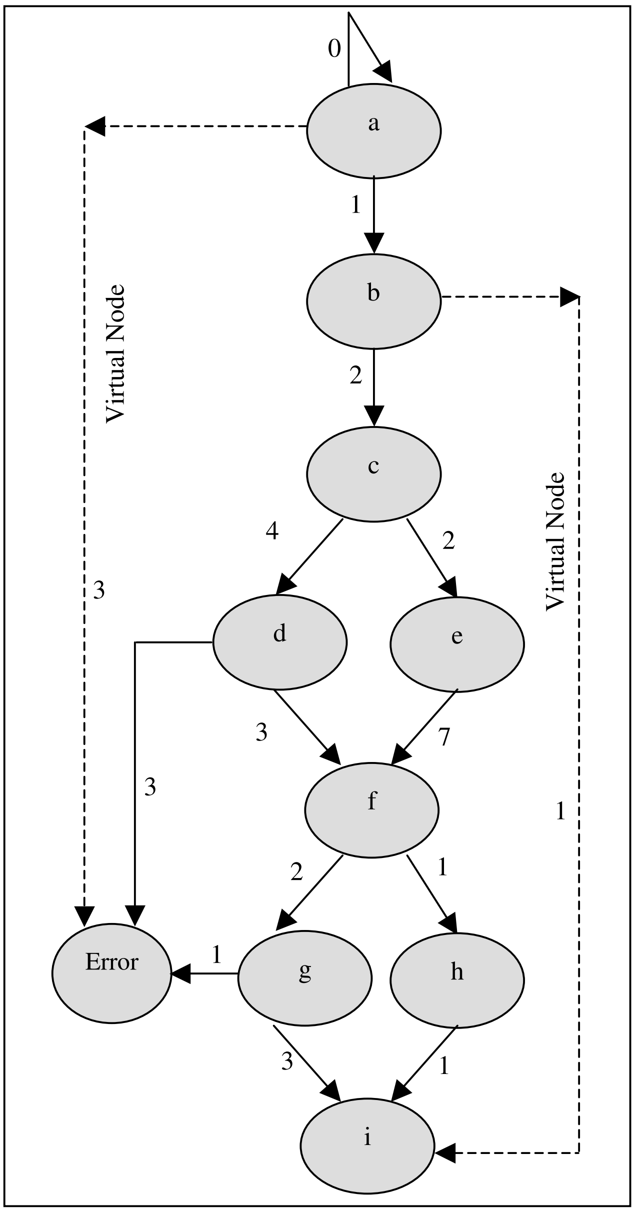

Fig. 5. Addition of an error and virtual nodes to UBMH.

	a	b	c	d	e	f	g	h	i	Sum
a	0	1	-	-	-	-	-	-	-	1
b	-	-	2	-	-	-	-	-	-	2
c	-	-	-	4	2	-	-	-	-	6
d	-	-	-	-	-	3	-	-	-	3
e	-	-	-	-	-	7	-	-	-	7
f	-	-	-	-	-	-	2	1	-	3
g	-	-	-	-	-	-	-	-	3	3
h	-	-	-	-	-	-	-	-	1	1
Sum	0	1	2	4	2	10	2	1	4	26

Table 1. Matrix of transition probabilities for OARS.

	a	b	c	d	e	f	g	h	i	Error	Sum
a	0	1	-	-	-	-	-	-	-	-	1
b	-	-	2	-	-	-	-	-	-	-	2
c	-	-	-	4	2	-	-	-	-	-	6
d	-	-	-	-	-	3	-	-	-	3	6
e	-	-	-	-	-	7	-	-	-	-	7
f	-	-	-	-	-	-	2	1	-	-	3
g	-	-	-	-	-	-	-	-	3	1	4
h	-	-	-	-	-	-	-	-	1	-	1
Sum	0	1	2	4	2	10	2	1	4	4	30

Table 2. Matrix of transition probabilities with error node for OARS.

	a	b	c	d	e	f	g	h	i	Error	Sum
a	0	1	-	-	-	-	-	-	2	3	6
b	-	-	2	-	-	-	-	-	-	-	2
c	-	-	-	4	2	-	-	-	-	-	6
d	-	-	-	-	-	3	-	-	-	3	6
e	-	-	-	-	-	7	-	-	-	-	7
f	-	-	-	-	-	-	2	1	-	-	3
g	-	-	-	-	-	-	-	-	3	1	4
h	-	-	-	-	-	-	-	-	1	-	1
Sum	0	1	2	4	2	10	2	1	6	7	35

Table 3. Matrix of transition probabilities with error and virtual nodes for OARS.

3.2 Failure analysis of UBMG

Now, we extend the UBMG to include the failure data. To capture the failure data, the access logs are scanned for HTTP return error codes of 4xx and 5xx as mentioned in [10]. Besides this, the errors from other servers are also considered. Theoretically, the error node can stem from any page in the graphical view. We add the error node 'Er' and all the page errors are associated with this node. The matrix of transition probabilities will have an additional column to represent the error node. A cell (m, Er) of this column will include the probability of transitioning from the node m to error node Er. Considering the OARS example, the view of UBMG with the addition of error node is shown in figure 4. The matrix of transition probabilities for the figure 4 is shown in Table 2. The matrix considers only those sessions that have some error. Of all the requests that enter node d, 35% of them encountered some error. Before proceeding to failure analysis due to service-level agreements (SLA) violation, we define the term Session Response Time (SRT) which is the sum of the service times of all the pages in the session. We define the SLA at session level and hence we need the desired response time target for each session. The access log files can be used to determine the page service time (PST) values. For example, in the IIS Web server the time-taken field represents the time spend by server to respond to the request. SRT is computed as the sum of PST's of its individual pages. Further, we compute the number of successful sessions where the SLA was violated. Let S1 and S2 be two sessions for the OARS example. Table 4 shows the sessions information, where each session is represented by a unique column, and includes number of successful sessions, number of instances of SLA violation, etc. The probability of reaching exit node for a session is computed as the ratio of number of exits with respect to the number of visits at the entry page. Figure 5 shows the addition of virtual nodes to the existing figure 4. The matrix of transition probabilities for the figure 5 is shown in Table 4.

Sessions	S1	S2
1. Total no. of successful session	125	150
2. Total no. of SLA violation $N_{SLA-FAIL}$	64	67
3. Probability of failures due to (2)	0.59	0.56
4. Prob. of reaching exit node for each session	0.78	0.76
5. Probability of SLA violation for each session using (3) and (4)	0.37	0.32

Table 4. Results of SLA violation probability.

3.2.1 Reliability Calculation

To compute the reliability of software code-level failures, we resort to determine the probability of encountering the failure node $P_{CODE-ERROR}$ represented in Figure 3.2. To solve this probability of reaching the error node, we formulate a set of equations from the matrix and use techniques like Cramer' s Rule, Matrix Inversion or Gauss Jordan Elimination method (for solving the sets of simultaneous equations). We also compute (a) the total number of failures due to invalid session $N_{INVALID-SESSION}$, and (b) number of instances where successful sessions did not meet SLA as $N_{SLA-FAIL}$. The probability of occurrence of invalid sessions is computed using (a). The probability of failure for a session due to (b) is computed by considering the total number of its successful sessions. In the OARS example, the probability of such failures is 0.59 in Session 1 and 0.56 in Session 2. The probability of a session reaching the exit node, but violating SLA or invalid sessions needs to be computed. The total session failure probability $P_{SESSION-FAILURE}$ is calculated as the sum of all the individual session probabilities and the probability of occurrence of invalid sessions. The overall probability of failure $P_{TOTAL-FAILURE}$ for the system is calculated as sum of the probability of reaching error node $P_{CODE-ERROR}$, and the probability of session failure $P_{SESSION-FAILURE}$ for the entire system. The overall reliability R_{SYSTEM} of the system is calculated by the equation (2):

$$R_{SYSTEM} = 1 - P_{TOTAL-FAILURE} \quad \textbf{(2)}$$

Thus the reliability computation is driven by failures at software code level, failures due to SLA violation and invalid sessions.

4. Web page trace algorithms

Consider a Web page trace W(n) = r (1)*r(2)*----*r(n) consisting of n Web page numbers (WPNs) requested in discrete time from 1 to n, where r(t) is the WPN requested at time t, and Web page trace is a sequence of Web page numbers. Therefore we define two references distances between the repeated occurrences of the same page in W(n). The forward distance '$f_t(x)$' for page 'x' is the number of time slots required from time 't' to the first repeated reference of Web page 'x' in the future:

$$f_t(x) = \begin{cases} \text{k, if k is the smallest integer such that r(t+k)=r(t)} \\ \text{=x in W(n)} \\ \infty\text{, if x does not reappear in W(n) beyond time t} \end{cases}$$

Similarly, we define a backward distance $b_t(x)$ as the number of time slots from time 't' to the most recent reference of Web page 'x' in the past:

$$f_t(x) = \begin{cases} \text{k, if k is the smallest integer such that r(t-k)=r(t)} \\ \text{=x in W(n)} \\ \infty\text{, if x never appeared in W(n) in the past} \end{cases}$$

Let R(t) be the resident set of all Web pages residing in Web server under the indexed Web page at time 't'.

4.1 Web page replacement policies

The following Web page replacement policies are specified in a Web server system for a Web Frame Repetition (WFR), which results in Web page faults at time 't' [11].

(a) Least recently used (LRU): This policy replaces the Web page in R(t) which has the longest backward distance or it will replace the Web page that has not been recently used for the longest period of time.

(b) Optimal (OPT) algorithm: This policy replaces the Web page in R(t) with the longest forward distance or replaces the Web page that will not be used for the longest period of time.

(c) First-in-first-out (FIFO): This policy replaces the Web page in R(t) which has been in memory for the longest time or when a Web page must be replaced, the oldest Web page is chosen.

(d) Least frequently used (LFU): This policy replaces the Web page in R(t) which has been least referenced in the past or it will replace the Web page that has the smallest count.

(e) Most frequently used (MFU): This policy replaces the Web page in R(t) which has been most referenced in the past or it will replace the Web page that has the largest count.

Consider a Web server system with two-level hierarchy: main memory M1 and disk memory M2. The number of Web page frame (WPF) is 3, labeled a, b, and c; and the number of pages in M2 is 11, 13, and 15 as presented in table 5, 6 and 7. For these experimental validations a sequence of random number has been generated. The sequence of Web page numbers so formed is the Web page trace. The following three Web page trace numbers have been taken to experiment with the Web page replacement policies: (i) 0 1 2 4 2 3 7 2 1 3 1; (ii) 0 1 2 4 2 3 7 3 2 1 1 3 1 (with an error node); and (ii) 0 1 2 3 2 4 2 3 7 3 2 1 1 3 1 (with error and virtual nodes). The results from figure 6 indicate the superiority of OPT policy over others. However, the OPT is very difficult to implement in practice. The LRU policy performs better than FIFO due to the locality of references. The MFU policy shows better results than LFU policy due to the most frequent usability index of Web pages. The MFU and LFU policies are useful to keep a counter of the number of Web page references that have been made to each page. From these, results, we realize that the LRU is generally better than FIFO, MFU and LFU. However, exceptions still exist due to the dependence on program behavior.

4.2 Usability index for Web pages

The usability index has been implemented by means of a set of suggestions (page links) dynamically generated on the basis of the active user session, which are used to personalize page requested. Typically, the Web usability is structured according to two components, performed off-line and on-line analysis with respect to the Web server activity. By analyzing the historical data (i.e. server access log files), the off-line component builds a knowledge base, which is used in the on-line phase to generate the personalized content. This content can be expressed in several forms, such as links to pages or advertisements considered of interest for the current user. User sessions are identified by means of cookies stored on the client side. Cookies contain the keys to identify the client sessions. For each Web page URL request has been generated. The Web server knowledge base updated according to the characteristics of the current session, and then generated. Presuming that interest in a Web page depends on its content and not on the order in which a Web page is visited during a session, the edge weight is computed as $W=N_{ij}/\max\{N_i, N_j\}$, where N_{ij} is the number of sessions containing both pages I and j, and Ni and Nj are the number of sessions containing only page i or j. Generally, LRU algorithm is applied. According to this algorithm, information about a Web page less recently accessed is replaced with that for a currently accessed Web page. Therefore, the system performance due to least recently used Web page (s) will enhance the Web site size and their performance.

	WPF	0	1	2	4	2	3	7	2	1	3	1	Hit Ratio
LRU	a	0	0	0	4	4	4	7	7	7	3	3	
	b		1	1	1	1	3	3	3	1	1	1	
	c			2	2	2	2	2	2	2	2	2	3/11
	WFR					*			*			*	
OPT	a	0	0	0	4	4	3	7	7	7	3	3	
	b		1	1	1	1	1	1	1	1	1	1	
	c			2	2	2	2	2	2	2	2	2	4/11
	WFR					*			*	*		*	
FIFO	a	0	0	0	4	4	4	4	2	2	2	2	
	b		1	1	1	1	3	3	3	1	1	1	
	c			2	2	2	2	7	7	7	3	3	2/11
	WFR					*						*	
LFU	a	0	0	0	0	0	0	0	0	0	0	0	
	b		1	1	1	1	1	1	1	1	1	1	
	c			2	4	2	3	7	2	2	3	3	2/11
	WFR									*		*	
MFU	a	0	0	0	4	4	4	4	2	2	2	2	
	b		1	1	1	1	3	3	3	1	1	1	
	c			2	2	2	2	7	7	7	3	3	2/11
	WFR					*						*	

Table 5. Results of Web page replecement policies for figure (3).

	WPF	0	1	2	4	2	3	7	3	2	1	1	3	1	Hit Ratio
LRU	a	0	0	0	4	4	4	7	7	7	1	1	1	1	
	b		1	1	1	1	3	3	3	3	3	3	3	3	
	c			2	2	2	2	2	2	2	2	2	2	1	5/11
	WFR					*			*	*		*	*		
OPT	a	0	0	0	4	4	3	3	3	3	3	3	3	1	
	b		1	1	1	1	1	1	7	7	1	1	1	1	
	c			2	2	2	2	2	2	2	2	2	2	2	5/11
	WFR					*			*	*		*	*		
FIFO	a	0	0	0	4	4	4	4	4	2	2	2	2	1	
	b		1	1	1	1	3	3	3	3	1	1	1	1	
	c			2	2	2	2	7	7	7	7	7	3	3	3/11
	WFR					*			*			*			
LFU	a	0	0	0	0	0	0	0	0	0	0	0	0	0	
	b		1	1	1	1	1	1	1	1	1	1	1	1	
	c			2	4	2	3	7	3	2	1	1	3	1	1/11
	WFR											*			
MFU	a	0	0	0	4	4	4	4	4	2	2	2	2	1	
	b		1	1	1	1	3	3	3	3	1	1	1	1	
	c			2	2	2	2	7	7	7	7	7	3	3	3/11
	WFR					*			*			*			

Table 6. Results of Web page replacement for figure (4) after addition of an error node

	WPF	0	1	2	3	2	4	2	3	7	3	2	1	1	3	1	Hit Ratio
LRU	a	0	0	0	3	3	3	2	3	7	3	3	3	3	3	3	5/11
	b		1	1	1	1	4	4	4	4	4	4	1	1	1	1	
	c			2	2	2	2	2	2	2	2	2	2	2	2	2	
	WFR					*						*	*		*	*	
OPT	a	0	0	0	3	3	3	3	3	3	3	3	3	1	1	1	5/11
	b		1	1	1	1	4	2	2	7	3	3	3	3	3	3	
	c			2	2	2	2	2	2	2	2	2	1	1	1	1	
	WFR					*			*			*			*	*	
FIFO	a	0	0	0	3	3	3	3	3	3	3	2	2	2	3	3	3/11
	b		1	1	1	1	4	4	4	4	4	4	1	1	1	1	
	c			2	2	2	2	2	2	7	7	7	7	1	1	1	
	WFR					*			*							*	
LFU	a	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2/11
	b		1	1	1	1	1	1	1	1	1	1	1	1	1	1	
	c			2	3	2	4	2	3	7	3	2	2	1	3	3	
	WFR												*			*	
MFU	a	0	0	0	3	3	3	3	3	3	3	2	2	2	3	3	2/11
	b		1	1	1	1	4	4	4	4	4	4	1	1	2	2	
	c			2	2	2	2	2	2	7	7	7	7	1	1	1	
	WFR								*							*	

Table 7. Results of Web page replacement for figure (5) after addition of an error and virtual nodes.

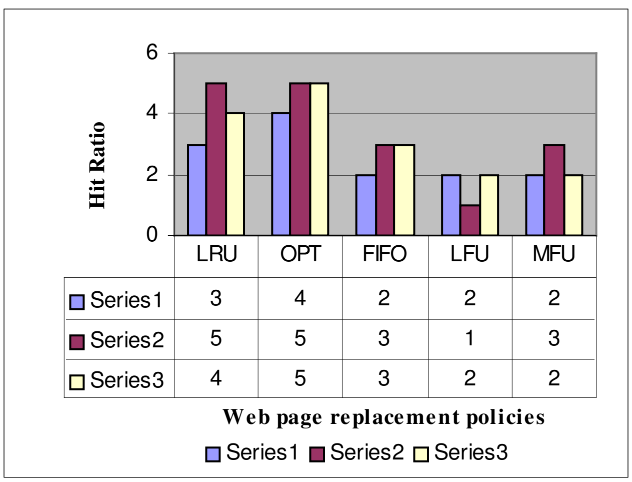

Fig. 6. Final characteristics of Web page replacement policies for table 5, 6 and 7.

4.3 Web page error estimating measure

We consider 'H' as hyperlink and 'I' as index page of a Web based application. The probability value P(HPA) refers to the probabilistic hyper-threading information between the presence or absence of H in an index page 'I'. The P(rel) indicates the probabilistic relation of 'H' for a given index page 'I'. Hence error-estimating function EE(I) is given by:

$$EE(I) = \log P(HPA/ rel) / P(HPA) \quad \textbf{(3)}$$

Where: P(HPA) is an approximate probability. If P(HPA) tends to be '1', then equation (3) becomes EE(I) = log P (HPA/rel). Since the hyperlinks frequently updates the concepts in case of dynamic or Active X pages. Therefore, the unique or monotonous results are not obtained during search. This leads a fact that the value of Web page under hyperlink Hi (where i=1,2,3,----,n) will change to another Hi, when reliability percentage of a Web page is concerned. The relation Hi and Hi can be compared at any instance based on the comparator rules such as (i) Hi = Hi; (ii) Hi ≈ Hi; (iii) Hi < Hi; and (iv) Hi > Hi. The rules indicate that the term is focused on utilization of hyperlinks.

5. The RSWAEA Method

In order to deal with the problem of effort estimation we have been studying the last two years the Web-based software development processes, related to the development of small and medium size Web-based information systems. Based on the analysis of these results, we identified a low usability of the well-known effort estimation methods and a necessity of a model to support estimation in such scenario. Due to this, we developed a method for fast estimating the Web-based software development effort and duration, which will definitely be adapted by the software community for the development of Web-based hyper media applications. We called it RS Web Application Effort Assessment (RSWAEA) method. The method will be very useful to estimate the development effort of small to large-size Web-based information systems. The DWOs (Data Web Objects) are an approximation of the whole size of the project; so, it is necessary to know what portion of the whole system DWOs represent. This knowledge is achieved through a relatively simple process (briefly described in next subsection). Assuming that the estimation factors in the computation of the effort are subjective, flexible and adjustable for each project, the role of the expert becomes very relevant. Once the value of the portion or representativeness is calculated, the expert can adjust the total number of DWOs and he/she can calculate the development effort using the following equation (4).

$$E = (DWO \,.\, (1+X^{*}))^{P} \,.\, CU \,.\, \prod_{i=1}^{8} cd_i \quad \textbf{(4)}$$

Where: E is the development effort measured in man-hours, CU is the cost of user, cd_i is the cost drivers, DWO corresponds to the Web application size in terms of data web objects, X* is the coefficient of DWO representativeness, and P is a constant. The estimated value of real data web objects (DWO) is calculated as the product of the initial DWOs and the representativeness coefficient *X**. This coefficient is a historical value that indicates the portion of the final product functionality that cannot be inferred from the system data model. The value of X* (coefficient of DWO representativeness) is between 1 to 1.3 depending upon small to large-size We-based applications. The process of defining such coefficient is presented in the next section. The cost of each user is has the values between 0 and 5. A value of CU of 0 means the system reuses all the functionality associated with each user type; so, the development effort will also be zero. On the other hand, if the cost of user is five, this means that there is no reuse of any kind to implement the system functionality for each user type. It represents the system functionality that is associated with each user type. The defined cost drivers (cdi) are similar to those defined by Reifer for WebMo [12]. The last adjustable coefficient in RSWAEA corresponds to constant P that is the exponent value of the DWO. This exponent is a value very close to 1.01, and it must neither be higher than 1.12 nor lower than 0.99. This constant's value depends on the project size measured in DWOs. In order to determine this value, various statistical analyses have been done on various Web-based applications. As a result, this constant was assigned the value 1.09 for projects smaller than 300 DWOs, and 1.03 for projects larger than 300 DWOs [13].

6. Conclusions and future work

In this paper we have introduced an approach for determining the reliability, usability index, error estimating function, Web replacement policies, UBMH, RSWAEA and effort estimation for Web-based systems. These method work by offline and online analysis of Web logs and come up with useful metrics like usability index, error estimating function, Web replacement policies, cost, RSWAEA, UBMG, session count, SRT computation etc., and these metrics can effectively be used for the computation of reliable efforts for small to larger-size Web-based applications. Although these methods do not replace the expert estimator, but they provide him/her with a tool for achieving a more accurate estimation, based on real data in a shorter time. Estimating the cost, usability, error, duration and reliability of Web developments has a number of challenges related to it. To handle these challenges, we have analyzed many findings drawn from the experienced and expert opinions. Finally, by taking the good qualities of a software metric and an accessible Web design, we validated that the proposed models have better effort predictive accuracy than any existing traditional methods. In near future the development of Web-based applications using an object-oriented frame work, component-based framework, parametric-based framework or project-level framework and hardware-software co-designs related framework for sensitivity analysis and risk identification need to be designed. Our future work may include the study of lexical analysis together with COTS to develop the complete framework for effort assessment for authoring Web-based applications. However, positive results would suggest that the various efforts applied to estimate Web-based applications, would be an invincible task for the upcoming future.

7. Acknowledgments

A major part of the research reported in this was carried out at U.I.E.T and D.C.S.A, K.U.K, Haryana, India. We are highly indebted and credited by gracious help from the Ernet section of K.U.K for their constant support and help while testing our proposed models in different environment. The authors would like to thank those nameless individuals who worked hard to supply the data.

8. References

[1] Pressman S. Roger, *Software Engineering- A Practitioner's Approach* (McGraw-Hill, 1997).

[2] *IEEE Trans. Software Engineering,* Vol. SE-10, pp-728-738, (1984).

[3] Ejiogu, L., *Software Engineering with Formal Metrics* (QED Publishing, 1991).

[4] Roche, J.M., Software Metrics & Measurement Principles, *Software Engineering Notes, ACM*, Vol. 19, no. 1, pp.76-85, 1994.

[5] Basili, V.R., & D.M.Weiss, A Methodology For Collecting Valid Software Engineering Data, *IEEE Software Engineering Standards*, Std. 610.12-1990, pp.47-48, 1993.

[6] Daniel A. Menasce, Virgilio A.F. Almeida, Scaling for E-Business Technologies, Models, Performance, and Capacity Planning (Prentice Hall PTR, pp. 49-59, 2000).

[7] Shubhashis Sengupta, Characterizing Web Workloads– a Transaction-Oriented View, IEEE/ IFIP 5th International Workshop on Distributed Computing (IWDC- 2003).

[8] Wen-Li Wang, Mei-Huei Tang, User-Oriented Reliability Modeling for a Web System, *14th International Symposium on Software Reliability Engineering (ISSRE)*, November 17 - 21, 2003.

[9] D.A. Menace, V.A.F. Almeida, R. Fonseca, M.A. Mendes, A Methodology for Workload Characterization of E-Commerce Sites, *Proceedings of 1st ACM conference on Electronic Commerce*, 99.

[10] The Internet Society, Request for Comments (RFC): 2616. Hypertext Transfer Protocol–HTTP/1.1, http://www.w3.org/Protocols/rfc2616/rfc2616.html.

[11] Kai Hwang, Advanced Computer Architecture Parallelism Scalability Programmability, TMH- New Delhi, 2004.

[12] D.J. Reifer, Web Development: Estimating Quick–to Market Software, IEEE Software, Vol. 17, No. 6,Pages 57 - 64, 2000.

[13] Sanjeev Dhawan, Rakesh Kumar, Effort Assessment and Predictions for High Volume Web-based Applications: a Pragmatic Approach, International ERCIM Workshop on Software Evolution 2006 (Université des Sciences et Technologies de Lille, France, pp. 57-64).