Automatic Authoring in the LAOS AHS Authoring Model

Alexandra I. Cristea
Faculty of Computer Science and Mathematics
Eindhoven University of Technology
Postbus 513, 5600 MB Eindhoven, The Netherlands
+31-40-247 4350
a.i.cristea@tue.nl

ABSTRACT

In this paper, we extend the automatic authoring techniques that can be built based on the LAOS model, a five-layer AHS authoring model. As the LAOS model itself is fairly complex, although information-rich, an adaptive hypermedia author needs a lot of system support to be able to populate all its levels with the corresponding information. Therefore, such automatic authoring techniques, which are actually automatic transformation (and interpretation) rules between the different layers of the model, have been designed. These automatic rules represent, in the area of adaptive systems, designer-goal oriented adaptation techniques. They should represent the goal of the designer that is authoring the hypermedia (such as the pedagogical goal in educational adaptive hypermedia). Therefore, this paper represents yet another step towards an adaptive hypermedia (or adaptive course) that 'writes itself'. The focus here is on automatic transformation between the *domain* and a newly introduced *goal and constraints model*, to show that the effort of introducing this new layer can be minimal.

Keywords

Adaptive authoring, adaptive hypermedia, AHS, AHAM, ontologies, MOT, Dexter model

1. INTRODUCTION

Adaptive hypermedia system (AHS) are becoming nowadays popular, due to their connection with the W3C and IEEE LTTF [18] movements towards (*ontology*-based) customization and the *semantic Web* [24]. The success of commercial adaptive systems as Firefly, or research AHS as AHA! [15], Interbook [4], TANGOW [6] and others has pushed AHS forward. Their edge over classical ITS systems [3] relies on their simplicity: they contain a simple domain -, user model (usually an overlay - of the domain model), aimed at a quick response, which is extremely beneficial in the speed-concerned WWW environment. However, for quite a long while there has been a lack of powerful authoring tools for adaptive hypermedia [2][7]. One of the main reasons was the great (but fruitful) diversity in AHS implementations, many with implicit models [27].

Here we build on AHAM [27] and on the LAOS model [12] that allows a more flexible model for adaptive hypermedia authoring. As authoring of information rich adaptive hypermedia is difficult and time-consuming, we have added, next to the LAOS model that allows high flexibility, some methods to bypass the workload for the adaptive hypermedia author. Here we show, for instance. (adaptive, adaptable) automatic authoring techniques that can lead to more powerful AHS authoring tools. Instead of having the author populate the layers of an adaptive hypermedia model such as LAOS, the system can take many of the tasks over and perform them with no or little authoring intervention. Here we are going to highlight some examples of such *automatic authoring*, as we call it.

The paper is organized as follows. In section 2 we briefly recall the LAOS model as well as the definitions we need for the automatic transformations. Section 3 introduces automatic transformations that can lead to designer adaptation: automatic adaptation of the designed hypermedia (e.g., courseware) itself to the designer goal. In section 4 we exemplify some automatic transformations between two concrete layers, the *domain* and the *goal and constraints layer*, that are allowed by the LAOS model, and compute some *flexibility degrees* to show the expressiveness of the possible transformations and give also some examples and implementation instances from MOT [13]. In section 5 we present a short discussions about the benefits and implications of such automatic, designer goal oriented transformations. Finally, section 6 draws conclusions.

2. LAOS LAYERED MODEL

The LAOS model (figure 1) is a generalized model for generic adaptive hypermedia authoring and was introduced in [12], therefore we are not going into many details about it here. It is based on the AHAM model [27], which in turn extends the Dexter reference model [17] for the specific field of adaptive hypermedia. The LAOS model is composed of five components: the *domain model* (DM), *goal and constraints model* (GM), *user model* (UM), *adaptation model* (AM) and *presentation model* (PM), as can be seen in Fig. 1.

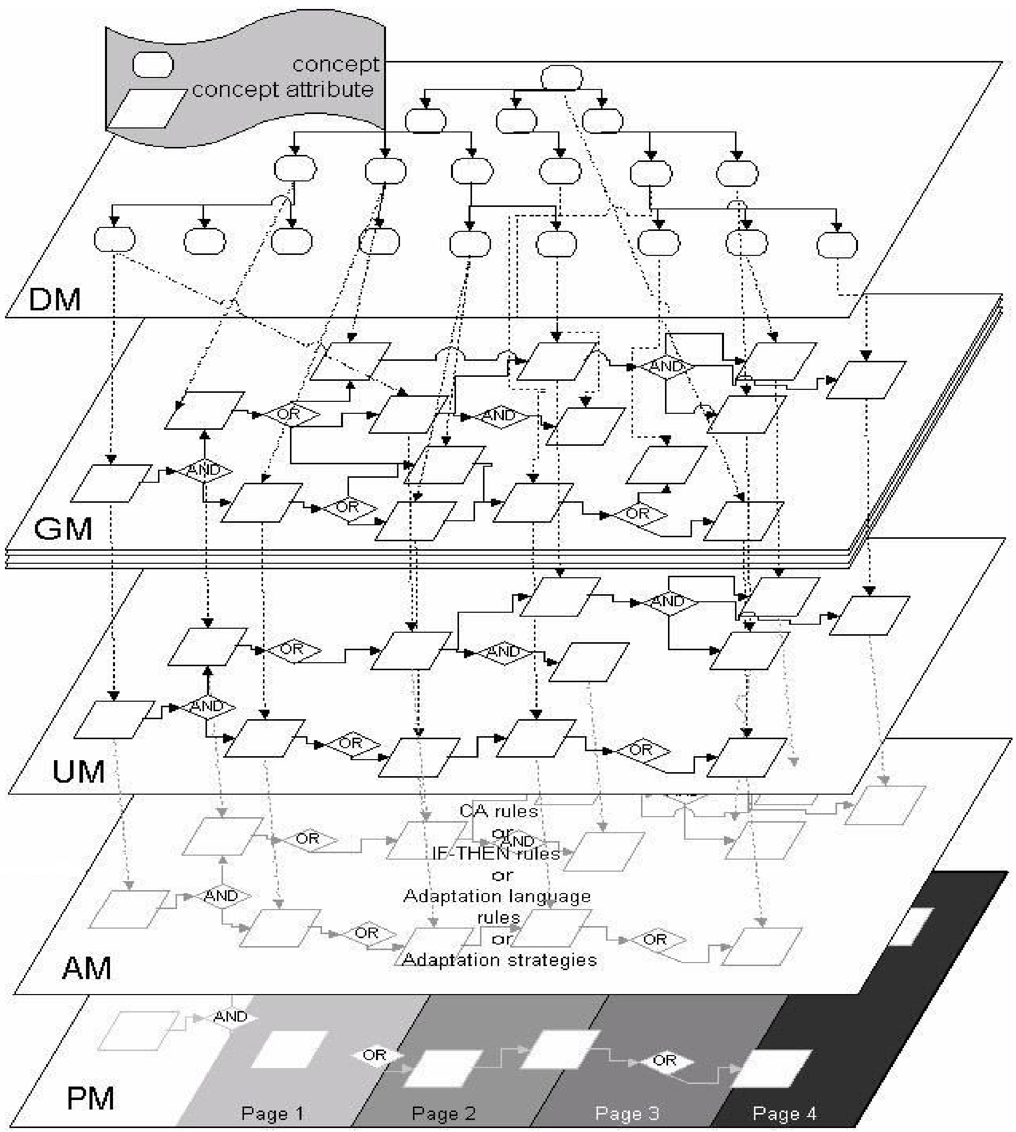

Figure 1. The five level AHS authoring model.

The idea is based on the book–course or book–presentation metaphor: generally speaking, when making a presentation (GM), be it for the Web or not, we base it on one or more references (DM). Simplifying, a presentation (GM) is based on one or more books (DM)[1]. This is why we need an intermediate presentation (GM) layer. The rest of the layers are shared with the AHAM model, so the motivation of using them is similar to the motivation of the previous model.

The basic idea is that such a model is easier to maintain than a small but compact model with all needed information in the same place. A change in user information will go, for instance, directly into the *user model* (and might influence the adaptation model) but has nothing to do with the domain – so will not influence the *domain* - or *goal and constraints model*. A presentation style change or update, on the other hand, will influence only the *goal and constraints model* (if it is a content related presentation style change) or only the *presentation model* (if it is a interface related change). So, each type of information is kept separately from information of other type, thus allowing maintainability.

Moreover, with the LAOS structure, dynamic (adaptive) presentation generation becomes possible. The actual presentation seen by the user can contain both elements of the *goal and constraints* - as well as elements of the *domain model* (e.g., for clarification of an explanation based on only the GM, the other elements/ objects of the respective concept, or the other concepts related to the current concept, can be referred, via a jump over one layer). This increases the flexibility and expressivity of the created adaptive presentations, as we shall see by computing the flexibility indexes of automatic transformations, for instance[2].

In the following we are going to list the definitions regarding the different layers that are going to be used in the automatic transformations.

2.1 Conceptual Layer

Definition 1. We consider a concept map *CM* of the AHS to be determined by the tuple <C,L>, where C represents the set of concepts and L the set of links ($CM \subseteq \mathsf{CM}$, the set of all concept maps of the AHS).

Definition 2. A concept $c \in C$ is defined by the tuple $< A_c, C_c >$ where A_c ($A_c \neq \emptyset$) is a set of attributes and C_c a set of sub-concepts.

Definition 3. A_{min} is the minimal set of (standard) attributes[3] required for each concept to have ($A_c \supseteq A_{min}$).

Definition 4. A concept $c \in C$ is a composite concept if $C_c \neq \emptyset$.

Definition 5. A concept $c \in C$ is an atomic concept if $C_c = \emptyset$.

Definition 6. A link $l \in L$ is a tuple $< c1, c2, n_l, w_l >$ with $c1 \in Ci$, $c2 \in Cj$ *start* and *end* concepts, respectively, n_l a name or label of the link and w_l a weight of the link.

[1] This is why the GM layer is so dense: from one DM multiple GM versions can be generated.

[2] Please note however, that automatic transformations represent in themselves a restriction on the total flexibility of the system, because they do not add new data, but are based on re-usage of inherent information. The actual flexibility allowed by the LAOS model (given by the combination of all possible elements allowed by LAOS) is therefore much greater.

[3] This is to 'force' the authors to give at least some minimal information about the concept they are defining, in order to be make the semantics of the concept machine-readable (minimal ontology-based meta-data tagging).

Definition 7. An attribute $a \in \mathbb{A}_c$ is a tuple <*var*,*val*>, where *var* is the name of the attribute (variable or type) and *val* is the value (contents) of the attribute[4].

Definition 8. Each concept *c* must be involved at least in one link *l*. This special relation is called *hierarchical link* (or link to father concept). Exception: root concept.

Concept *c* is determined by its identification $i \in \{1,\ldots,C\}$ (where $C=card(\mathbb{C})$) and the attributes of concept *i* are $a_i[h]$, with $h \in \{1,\ldots,A_i\}$ and $A_i \geq A_{min}$ (where $A_i=card(\mathbb{A}_c)$ and $A_{min}=card(\mathbb{A}_{min})$).

2.2 Goal and Constrains Layer

The goal map *GM* of the AHS is a special *CM*, as follows.

Definition 9. A concept $c \in \mathbb{C}$ in *GM* is defined by the tuple $< \mathbb{A}_c, C_c >$ where $\mathbb{A}_c$ ($card(\mathbb{A}_{min})=2$) [5] is a set of attributes and C_c a set of sub-concepts.

Definition 10. A link $l \in \mathbb{L}$ in *GM* is a tuple $<c1, c2, n_l, w_l>$ with $c1 \in \mathbb{C}$, $c2 \in$ CM.$\mathbb{C}$ [6] *start* and *end* concepts, respectively, n_l a name representing the type (i.e., hierarchical or AND/OR connections) of the link and w_l a weight of the link.

2.3 User Model

UM and AM have been described relatively well by AHAM [26].

However, another way of representing the UM is given in [10], where we view the UM as a concept map (CM). In such a way, relations between the variables within the UM can be explicitly expressed as relations in the UM, and do not have to be "hidden" among adaptation rules (figure 2). The components of the concept-based user model are:

1. a concept map of user variables and their values (UVM)
2. a history concept map (HM): an overlay model of visited attributes and concepts from the GM and DM (a copy or a pointer to them) with extra historical variables and their respective values attached (e.g., a set of [date of visit, duration] for each visit)
3. a future concept map (FM): an overlay model of attributes and concepts from the GM and DM to be visited (copy or pointers to them); this map should be in the general case dynamic, i.e., its components (concepts, attributes, links, variables, values) can vary according to the AM(adaptation model) application by the AE (adaptation engine) and the user's decisions.

---

[4] With values being volatile or not according to AHAM [26].

[5] Each *GM* concept has only 2 attributes: '*name*' and '*contents*'.

[6] Links can be added between any concept of the owned *GM* to any concept of the whole CM space of concepts, within *GM* or jumping a level, to the DM.

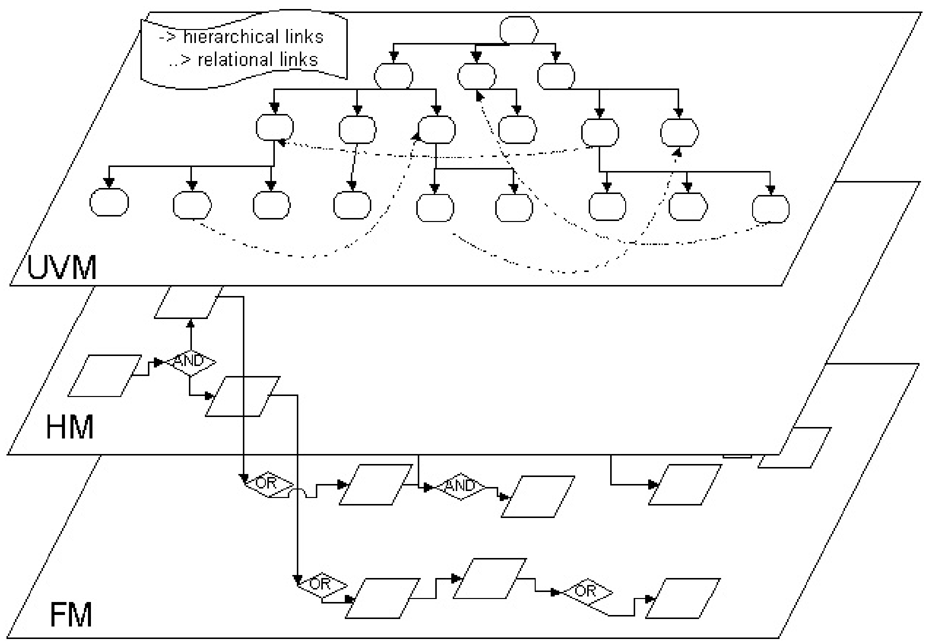

Figure 2: The 3-layered user model in LAOS.

2.4 Adaptation Model

We have introduced in [9] a new three-layer adaptation model (defining *low level assembly-like adaptation language*, *medium level programming adaptation language* and *adaptation strategies language*) that we are in the process of refining and populating, but this is beyond the scope of the present paper.

2.5 Presentation Model

The PM has to take into consideration the physical properties and the environment of the presentation and provide the bridge to the actual code generation for the different platforms (e.g., HTML, SMIL [25]).

3. ADAPTIVE, ADAPTABLE GENERATION OF AUTOMATIC TRANSFORMATIONS

In [11] we have defined the notion of designer adaptation (and adaptability), as *adaptation (*and *adaptability) to* the *design (authoring) goals*. In this paper we elaborate on the different possibilities of implementing such adaptation, based on the LAOS adaptive hypermedia authoring model introduced in [12].

In the following, we will present some automatic transformation possibilities from one layer to the other of the LAOS model, which can be performed in exclusivity by the authoring system, triggered or not by the designer's (author's) specific request.

These processes are normally done by hand during adaptive hypermedia design and authoring, and many of them are considered to be domain dependent (and therefore embedded in the domain functionality).

Here we find some patterns that allow us to generalize and automatically perform these transformations. In the first case, we talk about *adaptable design generation*, whereas in the second, we can talk about *adaptive design generation*. A simple rule system, for instance, can be implemented to make the choice between adaptability and adaptivity, and then within adaptivity, among the different adaptive options[7] presented.

Adaptivity implies that the system makes the inferences about the possible choices, and then takes the decision that is conforming to its adaptation model. The system then executes the choice.

In *adaptability*, the inferences about possible choices, as well as the selections are made by the user, and the system then executes the choice.

However, a combined version of adaptivity and adaptability is possible. The system can make the inferences about possible choices, and then allow the user to make the selection (or decision). This we call *adaptive adaptability*.

Moreover, we look at the *flexibility index* of many of the automatic transformations presented in the following sections, defined as the combinatorial index giving all the functionalities that can be covered by such transformations.

4. FROM DOMAIN MODEL TO GOAL AND CONSTRAINTS MODEL

This section discusses the automatic (adaptive, adaptable) *goal and constraints model* generation from the *domain model*, according to some presentation constraints and goals (e.g., pedagogical strategy or pedagogical technique). This transformation can be viewed as the first step from *information* to *knowledge*. This is due to the fact that, as said in [13] for instance, the *lesson*[8] *level* repeats the information contained in the concept level, now modeled and grouped based on pedagogical goals.

---

[7] here, transformations.

[8] an educational-oriented version of the LAOS goal and constraints level.

4.1 According to Concept Attribute Type

A very simple way of using the concept attributes can be for the selection of the specific types that are the only ones to go in the goal and constraints model. This transformation has been used for demonstration purposes by the MOT adaptive hypermedia authoring system [13].

I.e., for $\mathcal{A}_{min}$ ={title, keywords, introduction, text, explanation, pattern, conclusion} (A_{min} =7) as defined in section 2.1, we define $\mathcal{A}_{transf} \subseteq \mathcal{A}_{min}$ as $\mathcal{A}_{transf}$ ={title, introduction} (A_{transf} =2), as the transfer set from DM to GM, for implementing a goal-constraints model representing the elements for the pedagogical goal "introductory presentation" (e.g., for a beginners course, or for an overview on the whole material).

If $\mathcal{A}_{transf}$ ={title, explanation} (A_{transf} =2) we can implement a goal-constraints model representing the elements for the pedagogical goal "motivational presentation" (e.g., for a motivational overview on the whole material, that is to attract students towards it).

As a third example, we mention the selection of $\mathcal{A}_{transf}$ ={title, text} (A_{transf} =2), with which we can implement a goal-constraints model for a "motivational presentation" (e.g., for a motivational overview on the whole material, to attract students towards it).

As a fourth and last example, we mention the selection of $\mathcal{A}_{transf}$ ={title, keywords, pattern, conclusion} (A_{transf} =4) that implements a goal-constraints model representing the elements of the pedagogical goal "rehearsal" (e.g., for a summary or resuming presentation of the whole material, that is to remind students what they have learned in that lesson, and what the main important points – patterns - are).

Obviously, many more such pre-selections can be done. It is easier to imagine these types of transformations, if we go back to the book metaphor, and we are trying to construct a presentation based on a book: first, we are going to select the material that is going to be presented, as the whole book might be too long (hence, *constraints*) and its focus might be elsewhere (hence, *goal*). In this way, we build our *goal and constraints model*. Next, we are going to focus on the order, style, etc. of the presentation, which will appear in the *adaptation model* (and partially in the *presentation model*). Finally, we will interact with our audience and decide on skipping some parts or going into more details into others, depending on their reaction to our story (so, *user model* building and processing).

If we look at the combinatorics of these transformations, the *flexibility degree*, computing in this case the number of different lesson materials (so, the number of different sub-layers in the *goal and constraints model*) that can be generated automatically just with this simple procedure, is as follows. The different ways of selecting attributes from a concept C1 are:

$$flex(1) = \sum_{i=1}^{card(\mathcal{A}_{c1})} C(card(\mathcal{A}_{c1}), i) \geq \sum_{i=1}^{A_{min}} C(A_{min}, i) =$$
$$= \sum_{i=1}^{A_{min}} \frac{A_{min}!}{(A_{min}-i)!\,i!}$$

where *C(a,b)* are combinations of *a* elements taken *b* at a time.

This number is flex(1)≥87 for A_{min} =7 and represents the number of possible selections from C1 attributes if we don't care about the order. However, because in the *goal and constraints layer* the order starts being important, as opposed to the *domain layer*, the actual formula is:

$$flex(1) = \sum_{i=1}^{card(\mathcal{A}_{c1})} P(card(\mathcal{A}_{c1}), i) \geq \sum_{i=1}^{A_{min}} P(A_{min}, i) =$$
$$= \sum_{i=1}^{A_{min}} \frac{A_{min}!}{(A_{min} - i)!}$$

where *P(a,b)* are permutations of *a* elements taken *b* at a time.

So flex(1)≥ 13699, which is a much greater number.

Again, this is just the flexibility degree for one single concept and its extracted attributes. In a hypermedia concept map, there are many concepts. If we consider very simple automatic transformations, such as implemented in MOT [13], where all concepts in a concept map $\mathcal{C}$ are transformed in the same way, then these numbers don't change. However, if we allow concepts to be transformed independently, the flexibility degree will drastically grow.

4.2 According to Link Type

As said in the link definition for the conceptual layer (section 2.1), beside the obligatory hierarchical links, concepts can be involved in several other relations (links), which are defined by their *start point*, *end point*, *name* (type) and *weight*. In [11] we have shown that simply by using the attribute structure of the concepts, and labeling links between concepts with the name of the attribute that presents some relatedness, a great number of links can be automatically generated. However, in the LAOS structure we allow other types of links between concepts, which may not be automatically generated or related to attribute types.

These link types can be used to generate new, specific links at the level of the GM model.

A very simple example is the selection of some selected type of links only, that are to be taken over by the GM model (e.g., only *name* links).

In MOT, the automatic transformation functions described in section 4.1 go hand in hand with an automatic transformation into a standard, hierarchical, ordered link structure.

In other words, the selected attribute subset will keep (almost) the same *hierarchical structure* as its DM source: if a concept *C1* was a sub-concept of concept *C2*, and, let's say, we use the transformation of choosing only the {*title*, *introduction*} attributes; then, *L11=C1.title* and *L12=C1.introduction* will be sub-concepts of *L21=C2.title*, and the former attribute *C2.introduction* becomes concept *L22*, which is also a sub-concept of *L21*. In this way, the hierarchical link structure in *domain model* is translated into a hierarchical link structure of the *goal and constraints model*[9]:

$$LL21 \supseteq LL22, LL11, LL12$$

Moreover, concepts in the GM have an *order relation*, as opposed to concepts in the DM, which are represented as concept *sets* (so without order within a hierarchical depth). The solution implemented in MOT is to first list the (selected) attributes of a DM concept, and then the sub-concepts of the same concept. In our example case, this means that the order relationship is:

$$LL21 > LL22 > LL11 > LL12.$$

[9] which can be regarded also as a hierarchical inclusion relation.

Finally, relations in the GM have a type, which can be hierarchical, as describe before, or {AND/OR}. The latter are relations between elements from the same hierarchical depth. Automatically, all elements at a certain hierarchical depth are transformed in the MOT GM into concepts connected via an 'AND' relation:

AND(LL21 , AND(LL22 , LL11 , LL12)).

These can then be manually altered (e.g., into 'OR' relations), and added weights, but we are not going into details about this here.

The link-based transformation above is very simple, taking into account just the hierarchical link relations in the DM, but it is useful to illustrate the many different types of links that can be generated in the GM from only such a simple link sub-set. Here, one hierarchical relationship (together with the implicit attribute relationship) at DM level generates 3 relationships at the GM level. Please note that the above transformations don't take into account the relatedness links. By using these relations we could design an extended version of the three GM links above, as follows.

If, for instance, in the above setting, concept *C1* is related (e.g., via a 'text' attribute relatedness relation) with *C3* (*link(C1,C3,'text','70%')*), and we write the new GM concepts resulting as *LL31=C3.title* and *LL32=C3.introduction*, then we could write an automatic transformation from *domain* - to *goal and constraints model* that would generate:

LL21 $\supseteq$ LL22 , LL11, LL12,LL31,LL32

LL21 > LL22 > LL11,LL31 > LL12,LL32

AND(LL21 , AND(LL22 , LL11 , LL12,LL31,LL32)).

It is easy to see that this transformation would integrate in the introductory presentation also all related concepts.

4.3 Combination of Concept Attribute Type and Links

As previously said, MOT is already combining (a primitive version of) the above. However, much more complex and interesting combinations are possible.

The total number of possible combinations is obviously huge, as for each concept attribute type transformation there will be different possible link type transformations, making the total *flexibility degree* a *product of the independent flexibility degrees of the two transformation types*.

5. DISCUSSION

In sections 3, 4 we have shown a small, illustrative number of different types of automatic (adaptive, adaptable) transformation possibilities that can be directly performed by the adaptive hypermedia authoring system, in order to make the task of the author easier. These transformations are based on the data design defined by the LAOS model, which allows a concept-oriented approach for data design, analysis and usage.

It is interesting to note the great number of different design possibilities these automatic functions permit, computed in the form of a *flexibility degree*, which shows also the range of the adaptivity of the final system.

Moreover, although only some example transformations from the *domain* – to the *goal and constraints model* have been discussed and analyzed here, more types are possible (such as *domain* - to *adaptation model*, *goal and constraints* - to *adaptation model*, etc.). In practice it is reasonable to expect that these transformations will be parallel. This combination of all transformations may be leading to a situation where one transformation may be setting some restrictions on another one, but most of the time, these multiple transformations together will generate an increased number of possible functionalities.

We have not extended all the examples or computed the flexibility degree from all the cases, as the space in the paper did not permit it. Instead, we have tried to give some flavor of the different possible automatic transformations, their applicability and their diversity.

6. CONCLUSIONS

In this paper we have introduced different possible automatic authoring techniques between two specific layers of LAOS, a five level AHS authoring model with a clear-cut separation of the representation levels: the *domain model* (DM), the *goal and constrain model* (GM), the *user model* (UM), the *adaptation model* (AM) and finally the *presentation model* (PM).

We have previously shown [1] that authoring of adaptive hypermedia is a difficult task, which might be the main impediment that keeps AHS from being wider spread. Therefore we have implemented some goal-oriented automatic authoring techniques in MOT [13] that have the role to help the AHS author and ease the authoring burden. The implementation in MOT is mainly for demonstration purposes at this stage, and has therefore to be further developed.

In the current paper we have worked at the design for such a development from a more general, partially theoretical point of view. We have given a few examples for the automatic transformations, we have introduced and computed the flexibility degree offered by such transformations, and we have discussed the significance and extension possibilities of such transformations.

In this way, we are gradually advancing towards adaptive hypermedia that 'writes itself', being therefore adaptive not only to the final AHS user, but also to its designer (or author).

7. ACKNOWLEDGMENTS

This research is linked to the European Community Socrates Minerva project "Adaptivity and adaptability in ODL based on ICT" (project reference number 101144-CP-1-2002-NL-MINERVA-MPP).

8. REFERENCES

[1] Aroyo, L., Cristea, A.I., and Dicheva, D. A Layered Approach towards Domain Authoring Support. In Proceedings of ICAI 2002 (Las Vegas, US) CSREA Press.

[2] Brusilovsky, P. Adaptive hypermedia, User Modeling and User Adapted Interaction, Ten Year Anniversary Issue (Alfred Kobsa, ed.) 11 (1/2), 2002, 87-110.

[3] Brusilovsky, P., Schwarz, E., Weber, G. ELM-ART: An intelligent tutoring system on world wide web. In Proceedings of International Conference on Intelligent Tutoring Systems (ITS'96) (Montreal, Canada, June 1996), 261-269.

[4] Brusilovsky, P., Eklund, J., and Schwarz, E. Web-based education for all: A tool for developing adaptive courseware. Computer Networks and ISDN Systems, In Proceedings of Seventh International World Wide Web Conference (14-18 April 1998) 30 (1-7), 291-300.

[5] Calvi, L., and Cristea, A.I. Towards Generic Adaptive Systems Analysis of a Case Study. In Proceedings of AH'02 (Malaga, Spain, May 2002) Adaptive Hypermedia and Adaptive Web-Based Systems, LNCS 2347, Springer, 79-89.

[6] Carro, R. M., Pulido, E. Rodríguez, P. Designing Adaptive Web-based Courses with TANGOW .In proceedings of the 7th International Conference on Computers in Education, ICCE'99 (Chiba, Japan, November 4 - 7, 1999) V. 2, 697-704.

[7] Cristea, A.I., and Aroyo, L. Adaptive Authoring of Adaptive Educational Hypermedia, In Proceedings of AH 2002, Adaptive Hypermedia and Adaptive Web-Based Systems, LNCS 2347, Springer, 122-132.

[8] Cristea, A.I., and Calvi, L. The three Layers of Adaptation Granularity. UM'03. Springer (in press).

[9] Cristea, A.I., and De Bra, P. Towards Adaptable and Adaptive ODL Environments. In Proceedings of AACE E-Learn'02 (Montreal, Canada, October 2002), 232-239.

[10] Cristea, A.I., and Kinshuk. Considerations on LAOS, LAG and their Integration in MOT. ED-MEDIA'03 (in press).

[11] Cristea, A., De Mooij, A. Designer Adaptation in Adaptive Hypermedia. In Proceedings of ITCC'03 (Las Vegas, US 28-30 April 2003) IEEE Computer Society (in press).

[12] Cristea, A., De Mooij, A. LAOS: Layered WWW AHS Authoring Model and its corresponding Algebraic Operators. In Proceedings of WWW'03, Alternate Education track. (Budapest, Hungary 20-24 May 2003).ACM (in press).

[13] Cristea, A., De Mooij, A. Adaptive Course Authoring: MOT, My Online Teacher. In Proceedings of ICT-2003, IEEE LTTF International Conference on Telecommunications, "Telecommunications + Education" Workshop (Feb 23 - March 1, 2003 Tahiti Island in Papetee - French Polynesia) (in press).

[14] Cristea, A.I., Okamoto, T., and Kayama, M. Considerations for Building a Common Platform for Cooperative & Collaborative Authoring Environments. In Proceedings of AACE E-Learn'02 (Montreal, Canada, October 2002), 224-231.

[15] De Bra, P. and Calvi, L. AHA! An open Adaptive Hypermedia Architecture. The New Review of Hypermedia and Multimedia, vol. 4, Taylor Graham Publishers, 1998,115-139.

[16] European Community Socrates-Minerva project (project reference number 101144-CP-1-2002-NL-MINERVA-MPP). http://wwwis.win.tue.nl/~alex/HTML/Minerva/index.html

[17] Halasz, F., Schwartz, M. The Dexter Hypertext Reference Model. Communications of the ACM, Vol. 37, nr. 2, pp. 30-39, 1994.

[18] IEEE LTTF, Learning Technology Task Force. http://lttf.ieee.org/

[19] IMS (Instructional Management System) http://www.imsproject.org

[20] Kayama, M., Okamoto, T. and Cristea, A.I. Exploratory Activity Support Based on a Semantic Feature Map. In Proceedings of AH'00, LNCS 1892, Springer, 347-350.

[21] Mizoguchi, R., Bourdeau, J. Using Ontological Engineering to Overcome Common AI-ED Problems, International Journal of AI in Education, 11 (2), 107-121.

[22] My Online Teacher. http://wwwis.win.tue.nl/~alex/MOT01/TeachersSite-html/index.html

[23] Wu, H., De Bra, P. Sufficient Conditions for Well-Behaved Adaptive Hypermedia Systems. In Proceedings of the First Asia-Pacific Conference on Web Intelligence: Research and Development (Maebashi, October 2001). Lecture Notes in Artificial Intelligence, Vol. 2198, Springer, 148-152.

[24] WC3, Semantic Web. http://www.w3.org/2001/sw/

[25] W3C, SMIL, Synchronized Multimedia Language. http://www.w3.org/AudioVideo/

[26] Wu, H., De Kort, E., De Bra, P. Design Issues for General-Purpose Adaptive Hypermedia Systems. In Proceedings of the ACM Conference on Hypertext and Hypermedia (Aarhus, Denmark, August 2001) 141-150.

[27] Wu, H. A Reference Architecture for Adaptive Hypermedia Applications, doctoral thesis, Eindhoven University of Technology, The Netherlands, ISBN 90-386-0572-2.