Flexibility of Automatic Authoring for the Semantic Web

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

Abstract

The LAOS model, a 5-layer adaptive hypermedia (AH) authoring model, was previously shown to specify a flexible framework for (collaborative) creation of material for the semantic web. However, for adaptive behavior, an author has to design not only basic semantic contents (and its alternatives), but also specify the desired dynamics of the system, which is rather cumbersome. Therefore, automatic authoring techniques are being researched, that aim at decreasing the authoring burden. Here we elaborate on these techniques that can be built based on LAOS, and show specific implementations. They exploit the LAOS structure and consist of automatic transformation (interpretation) rules between different layers of the model (populate some layers based on the contents of others). To evaluate the effectiveness of these transformations, we have to see if and how much flexibility is lost by performing these automatic transformations, as opposed to fully manual creation. We shall see that even with these automatic transformations in LAOS, high flexibility can still be achieved.

1. Introduction

By embracing the goals of W3C and IEEE LTTF [16] communities towards (*ontology*-based [20]) personalization and the *semantic Web* [23], adaptive hypermedia systems (AHS) are gaining nowadays more popularity in different communities. They respond to one of the main goals of the semantic web, that to allow automatic semantic processing on the web. In the case of adaptive hypermedia (AH), this is achieved by adding the necessary data for 'intelligent' adaptive processing of web information. Mainly, this consist of contents alternatives and user model data, which specify which of the alternatives is relevant for which (type of) user. Successful research AHS such as AHA! [14], Interbook [5], TANGOW [6], but also commercial adaptive systems, such as Firefly, have proven the various benefits and customization variants of AHS. One of the big hindrances that stop the wider acceptance of AHS is the lack of powerful authoring tools [3]. Nowadays, the importance of authoring research for AHS is becoming clearer to the research community [2], and various ways to simplify the authoring process are sought. In this paper we show how, next to regular authoring in the LAOS model [13], (adaptive, adaptable) novel automatic authoring methods can be used for easier, more powerful AHS authoring [10]. In this way we illustrate how the LAOS model supports semantic web techniques. Some of these automatic techniques have been tested in practice [12], this being beyond the scope of this paper. The remainder of the paper is organized as follows. In section 2 we sketch the LAOS model. Section 3 elaborates on automatic transformations and machine interpretation of the information allowed by the LAOS model; we compute *flexibility degrees* and give concrete examples from MOT. Section 4 summarizes our contributions.

2. LAOS Layered Model

The LAOS model (figure 1, [13]), is a generalized model for *dynamic adaptive hypermedia authoring*, based on the AHAM model [26]. The model comprises five layers: the *domain model* (DM), *goal and constraints model* (GM), *user model* (UM), *adaptation model* (AM) and *presentation model* (PM). The revised and extended version of these components is shortly listed in the following subsections. From a semantic point of view, these layers represent ontologies, with exception of AM which specifies the interpretation and behavior of the elements of the ontologies. Their definitions are used for the explanation of the *automatic transformations* and should be best used as reference for section 3. By populating these ontologies, authors can create adaptive hypermedia for the semantic web.

Table 1. Generic Definitions.

Definition 1. Let **CM** be the set of all **AHS concept maps**.

2.1. Domain Model (DM)

The DM contains the basic concepts[1] of the contents and their representation, in the form of concept attributes[2].

[1] Concepts in LAOS have to have a *semantic* unity.

Next, we show domain model components definitions.

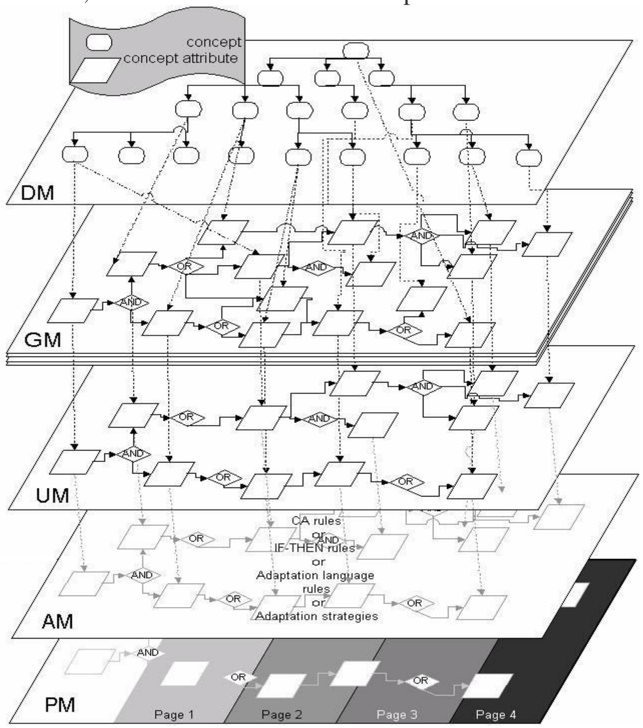

Figure 1. The five level AHS authoring model.

Table 2. Domain Model Definitions.

Definition 2. Let **DM** be the **set of all domain maps** (**DM** $\subseteq$**CM**), containing all information (resources and links between them) of the AHS relevant to the domain of the resources.
Definition 3. A **domain map** DM of the AHS is determined by the tuple <*C*, *L*, *Att*>; where *C* a set of concepts; *L* a set of links and *Att* a set of DM attributes[3] (DM $\in$ **DM**).
Definition 4. A **domain concept** $c \in \mathrm{DM}_i.C$ is defined by the tuple <*A*, *C*>; where $A \neq \emptyset$ is a set of DM attributes; *C* a set of DM sub-concepts; DM_i the domain map instance the concept belongs to.
Definition 5. A **domain link** $l \in L$ is a tuple <*S*, *E*, *N*, *W*> with *S*,*E* $\subseteq \{\mathrm{DM}_i.c_k\}_{i,k}$ ($S,E \neq \emptyset$) *start* and *end* sets of DM concept instances[4], respectively; *N* set of link labels; *W* set of link weights.
Definition 6. A **domain attribute** $a \in \mathrm{DM}_i.C.A$ is a tuple <*type*, *val*>, where *type* is the name of the DM attribute; *val* is the value (contents) of the DM attribute.
Constraint 1. Each domain concept is required to have a **minimal set of (standard) attributes**[5], A_{min} ($A \supseteq A_{min} \neq \emptyset$).

[2] Attributes in LAOS represent different aspects (views) about the same concept; e.g., a *title* is also an aspect of a concept. These attributes can be specified by any standard (e.g. IMS [17], LOM [19], SCORM [22], etc.) or be designer-defined attributes.
[3] Note that these are attributes at the level of the domain map, describing it directly, and not the concepts in it.
[4] c_k is a concept instance in an arbitrary domain DM_i. Please note that the generic definition allows loop links between a concept and itself. In praxis, links can be added between any concepts of the owned domain maps to any concepts of the whole space of domain concept maps.

Constraint 2. Each domain concept *c* must be involved in at least one special link *l*, called **hierarchical link** (link to ancestor concept). Exception: root concept.

2.2. Goal and Constraints Model (GM)

The GM is a constrained version of the domain model (DM) above, with *constraints* based on a *goal*. The idea is taken from the book–presentation metaphor: when designing a presentation (GM), we usually base it on some reference(s) (DM). For instance, a presentation (GM) can be based on one or more books (DM)[6]. The GM therefore already gives the presentation a preliminary shape. The actual presentation seen by the LAOS user however can still contain not only GM but also DM elements (e.g., for more information about a concept from the GM, other attributes of the respective DM concept, or other DM concepts related to it can be referred). This latter fact actually increases the flexibility and semantic expressivity of the created adaptive presentations, as we shall see, but, more importantly, separates links based on *content relatedness* (DM) from links based on *presentation structure* (GM). Following are the definitions for the components of the goal and constraints model. The GM is defined analogous to the DM, so the **GM** set of goal and constraints maps, GM map, *gl* goal link and *ga* attribute definitions are skipped here.

Table 3. Goal and Constraints Model Definitions.

Definition 7. A **goal and constraints concept** $g \in \mathrm{GM}_i.G$ is defined by the tuple <*GA*, *G*, $\mathrm{DM}_j.c.a$> $GA \neq \emptyset$ is a set of attributes; *G* a set of sub-concepts; $\mathrm{DM}_j.c \in C$ is the ancestor DM concept[7] and $\mathrm{DM}_j.c.a \in A$ is an attribute of that concept; GM_i is the name of the GM map instance to whom it belongs.
Constraint 3. Each goal and constraints concept *g* must be involved in at least one special *gl*, link called **prerequisite link** (link to ancestor concept) [8]. Exception: root concept.

2.3. User Model (UM)

UM can be a pure overlay model, as in AHAM [25], over the DM and GM previously defined. Another possible approach is to represent the user model [11] as a concept map, so that relations are also allowed. The UM is

[5] To specify what we REALLY want the authors to fill in.
[6] This is why the GM layer is so dense: from one DM, multiple GM versions can be generated. Similarly, for one presentation, several books can be used, so the GM-DM relation is a *multi-multi relation*.
[7] Can be void.
[8] GM concepts are also expected to participate in one of the special links called **AND/OR link** (link to sibling concepts), but as there is no constraint requiring the number of siblings to be above zero, this cannot be mentioned as a constraint.

defined analogous to the DM, so the **UM** set of user maps, UM user map, *ul* user link and *ua* user attribute definitions are skipped here.

Table 4. User Model Definitions.

Definition 8. A **user concept** $u \in UM_i.U$ is defined by the tuple <AU, U, $GM_i.g$ / $DM_j.c$>; $AU \neq \emptyset$ is a set of UM attributes; $UM_i.U$ a set of UM sub-concepts; $GM_i.g/DM_j.c \in G/C$ is the ancestor GM (or DM) concept.

2.4. Adaptation Model (AM)

In [10] we described a new, three-layer adaptation model (featuring: low level *assembly adaptation language*, medium level programming *adaptation language* and *adaptation strategies*) that we are in the process of refining and populating; this however, is beyond the scope of the present paper.

2.5. Presentation Model (PM)

The PM reflects only the physical properties and environment of the presentation; the PM provides the bridge to the actual code generation for the different platforms (e.g., HTML, SMIL [24]).
The PM is defined analogous to the PM, so the **PM** set of presentation maps, PM presentation map, *pl* presentation link and *pa* presentation attribute definitions are skipped here.

Table 5. Presentation Model Definition.

Definition 9. A **presentation concept** $p \in PM_i.P$ is defined by the tuple <PA, P, $GM_i.g$ / $DM_j.c$>; $PA \neq \emptyset$ is a set of PM attributes; $PM_i.P$ a set of PM sub-concepts; $GM_i.g/DM_j.c \in G/C$ is the ancestor GM (or CM) concept.

3. Adaptive, Adaptable Automatic Transformations

As well known, to create material for the semantic web is more difficult than creating material for the web of the past, because of the machine processable extra information needed. Even more so is the case when creating machine readable semantic information for the *adaptive* semantic web. Therefore, we look at automatic generation of some of the LAOS layers, based on information from others. Moreover, we also look at the *flexibility index* for these automatic transformations, to be able to measure how semantically expressive and computationally flexible these automatic generations are, as opposed to manual population of the layers.

Table 6. Flexibility Index Definition.

Definition 10. The **flexibility index** is the combinatorial index computing the number of different outcomes that can be generated by a given transformation.

3.1. From Domain Model to itself (DM→DM)

This section discusses the automatic (adaptive, adaptable) DM enrichment, according to its existing structure and contents. This means that implicit information contained in the DM is made explicit, with some information retrieval technique. We have already treated some specific DM to DM technique in [8], so this section will only shortly resume those results and extend them.

3.1.1. DM→DM: by concept attribute type. The easiest way to enrich the domain model is by finding automatically new links between existing concepts[9].
In [8] we have developed formulas for *relatedness relations* generation, for relations between concepts that share a common topic. This commonality was computed at concept attribute level, and therefore could automatically been given a type that corresponded to the (name of the) attribute type. In short, we could describe[10] the domain links[11] found by these computations as following:

```
If ∃c1∈DM1.C1, ∃c2∈DM2.C2 (DM1,DM2∈DM),
c1∈C1, c2∈C2, two domain concepts from two
possibly different domain concept maps;
c1=<A1,Cc1>, c2=<A2,Cc2>; ∃a1∈A1, ∃a2∈A2
two respective attributes of these
concepts, a1=<var1,val1>, a2=<var2,val2>.
If var1==var2 (the attributes have the
same type) a domain link can be generated
l=<{c1},{c2},{var1},{weight}> with
weight>0 defined as:
weight=number_common_features(val1,val2).
```

[12] gives different implementations of the function that computes the number of common features.

Table 6. Flexibility Index Definition.

Definition 11. The **mixed link flexibility index** is the number of possible (bidirectional) links of mixed type that can be generated between a selected set of concepts.

The *mixed flexibility index* of the links that can be generated between concepts *c1* (current concept) and *c2* is as follows (with shorthand notation Ai=card(*Ai*)):

[9] Please note that these new links can be between the concepts of the *current content* (concept map: e.g., course), between the current content and some *other content created by the same author*, or finally between the current content and some *other content created by a different author*.
[10] Notations are from the definitions in section 2.
[11] This is only one of the possible ways to connect concepts – stronger versions would look at ontological structures.

$mixflex(1,2) = A_1 A_2 \geq A_{\min}{}^2$;

If we want to consider links that have an unequivocal type[12], we obtain with the above notations the following *flexibility index* formula:

$$flex(1,2) = card(\mathcal{A}_{c1} \cap \mathcal{A}_{c2}) \geq card(\mathcal{A}_{\min}) = A_{\min}$$

where A_{ci} is the set of attributes of concept *i* and A_{min} is the minimal set of obligatory attributes, as previously defined.

If we consider we have C=card(C) concepts in the domain map DM, then the *flexibility index* between concept *c1* and the rest of the concepts in *C* is given by:

$$flex(1,*) = \sum_{j=2}^{C} card(\mathcal{A}_{c1} \cap \mathcal{A}_{cj}) \geq (C-1)A_{\min}$$

The *mixed flexibility index* between concept *c1* and the rest of the concepts would be:

$$mixflex(1,*) = A_1 \sum_{j=2}^{C} A_j \geq (C-1)A_{\min}{}^2$$

Generally, the *flexibility index* of concept map DM is given by the following relation:

$$flex(*,*) = \sum_{i=1}^{C} \sum_{j=i+1}^{C} card(\mathcal{A}_{ci} \cap \mathcal{A}_{cj}) \geq$$
$$\geq \sum_{i=1}^{C} \sum_{j=i+1}^{C} A_{\min} = \frac{C(C-1)}{2} A_{\min}$$

Similarly, the *mixed flexibility index* of concept map DM is:

$$mixflex(*,*) = \sum_{i=1}^{C} A_i \sum_{j=i+1}^{C} A_j \geq \frac{C(C-1)}{2} A_{\min}{}^2$$

Example 1: To give a concrete example, in the MOT adaptive hypermedia authoring system, A_{min} ={title, keywords, introduction, text, explanation, pattern, conclusion} so A_{min}*=7; in the concept map called 'Neural Networks I'* C*=card(C)=145, so flex(*,*)≥10440*7= 73080 and mixflex(*,*)≥10440*49= 511560.*

Please note that these are the connections implied by only one concept map. MOT already allows inter-linking of concept maps, which increases this number even more. Therefore, it is obvious that a great number of connections can be generated automatically, in this way making the adaptive hypermedia process easier, while adapting towards the authoring goal.

This is an explicit, symbolic way of linking concepts. However, this is not the only way of automatically finding concept links. Some years ago, in a different research group, we developed a sub-symbolic technique for concept clustering, based on SOM networks [18]. This clustering around topics can be combined with specific level-operators, as defined in [10], to write (student) user adaptation rules of the form:

IF *ENOUGH (L{V (c) | c∈ topic cluster})*
THEN *NEXT(topic)*

Where *c* is a concept in a concept list, L is the List operator and V is the View operator (as defined in [13]).

Therefore, different ways of automatically creating more expressiveness within the existing domain are possible, and there is space for more research in this direction.

3.1.2. DM→DM: by Link Type. By having an algorithm to check the link types, it may be possible (and beneficial) to create new links.

However, the most important contribution of link analysis would be to compare similar concepts[13] and to find that some attributes (or even sub-concepts) are missing.

Example 2: For instance, the concept called 'Discrete Neuron Perceptrons' from a Neural Networks course has an 'Example' attribute, whereas the concept 'Continuous Neuron Perceptrons' doesn't, although they are linked via their 'Title' attribute with a weight of 67%.

In such a case, the system could look for possible examples via other links to this concept, or signal the author about the possibly missing content item (attribute, sub-concept, etc.).

The *flexibility index*[14] for this link-based concept attribute retrieval from the link properties between the given concept *c1* (current concept) and some other concept *c2* can be defined as:

$$flex(1,2) = card(\mathcal{A}_{c2} - \mathcal{A}_{c1}) \geq 0$$

If we look at all the possible connections to *c1*, we obtain:

$$flex(1,*) = \sum_{j=2}^{C} card(\mathcal{A}_{cj} - \mathcal{A}_{c1}) \geq 0$$

Finally, for a whole concept map DM, the *flexibility index* is:

$$flex(*,*) = \sum_{i=1}^{C} \sum_{j=i+1}^{C} card(\mathcal{A}_{cj} - \mathcal{A}_{ci}) \geq 0$$

Depending on the variations in attribute design between the different concepts, this value can be large or can be zero.

Please note that an extended version of the content search could look *outside* the space defined by the LAOS model, such as the transition from a search within a closed space to the Web space.

[12] meaning that the attributes that determine the link are of the same type in both concepts, as stated by the link definition.

[13] Similar from a link-point of view, such as concepts sharing the same ancestor-concept, e.g., or concepts at the same level of the hierarchy, or concepts related with each other via some special link (of a given type), etc.

[14] For simplicity, we use the same notation for this link-based flexibility index, as we used for the concept-based index, although they obviously represent different values. Here, this number represents the number of potentially missing items (attributes).

3.2. From Domain - to Goal and Constraints Model (DM→GM)

Here we look at automatic (adaptive, adaptable) GM generation from the DM, according to presentation constraints and goals (e.g., for educational purposes we can envision pedagogical strategies or pedagogical techniques). This transformation represents the first step from *information* to *knowledge*. This was better detailed in [9], here are the essentials only, as follows.

3.2.1. DM→GM: by Concept Attribute Type. Concept attributes can be grouped into types that determine a filter for the selection of the items that will appear in the goal and constraints model.

Example 4: E.g., for A_{min} ={title, keywords, introduction, text, explanation, pattern, conclusion} (A_{min}*=7) as in section 2, if we define $A_{transf} \subseteq A_{min}$ as A_{transf} ={title, keywords} (*A_{transf}*=2), the transfer set from DM to GM, we can implement a goal-constraints model representing the elements for the pedagogical goal "short introduction" (e.g., for a very quick overview of the whole material).*
Example 5: If A_{transf} ={title, pattern} (A_{transf}*=2) we obtain a goal-constraints model representing the elements for the pedagogical goal "structural presentation" (e.g., for a review of the course).*

The *flexibility degree* that can be generated (showing the different ways of selecting attributes from a concept *c1*, considering that in the goal and constraints layer the order of concepts is important, as opposed to the domain layer), is as follows:

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

where *P(a,b)* are permutations of *a* elements taken *b* at a time. So, the flexibility degree for one single concept and its extracted attributes is flex(1)≥ 13699 (for A_{min} =7). If concepts are transformed independently, e.g., in special groups, this flexibility degree can grow significantly.

3.2.2. DM→GM: by Link Type. Links in the domain layer are defined (section 2.1) as either hierarchical, or of other nature. These link types can be used to generate specific links at the level of the GM model.
The simplest way is to select for the GM model only links of a specific type (e.g., only hierarchical links). In MOT, automatic transformations of hierarchical links are used to create a hierarchical, ordered link structure; i.e., the selected attribute subset will keep the same *hierarchical structure* as its DM source.
Example 6: If a concept ca was a sub-concept of concept cb in the DM and we use a similar transformation as in the previous subsection, of choosing this time the {title, text} attributes (A_{transf}=2); then, the generated La1=ca.title and La2=ca.text will be sub-concepts of Lb1=cb.title, and the former attribute cb.text becomes concept Lb2, which is also a sub-concept of Lb1. Therefore, the hierarchical link structure in DM is transformed into a new hierarchical link structure for the GM[15]: Lb1 ⊇ Lb2, La1, La2.
Furthermore, concepts in the GM are ordered, as opposed to concepts in the DM: Lb1 > Lb2 > La1 > La2.
Moreover, relations in the GM are typed; they can be hierarchical, as describe before, or {AND/OR}. The latter are relations between elements at the same hierarchical depth. In the MOT GM, all elements at a certain hierarchical depth are automatically transformed into concepts connected via an 'AND' relation. However, this can then be manually altered[16]:
AND(Lb1 , AND(Lb2 , La1 , La2)).
The illustrated link-based transformation above is simple, as it takes into account just the hierarchical link relations in the DM; however, it is useful in order to illustrate the many different types of links that can be generated for the GM from even such a simple link sub-set.

3.3. From Domain - to Adaptation Model (DM→AM)

This transformation represents automatic (adaptive, adaptable) AM generation from the DM, according to the (goal, e.g., pedagogical) strategy. The adaptation model has the role to interpret the other models: the domain -, goal – and even presentation model. Moreover, it can update these models and generate the presentation. In [10] we have defined the low-level adaptation (direct adaptation techniques) as:

a : {DM, GM, UM, AM, PM} →
{[DM], [GM], UM, [AM], PM}

Function *a* can furthermore be divided into a set of sub-functions: *a = {update, generate}* where:

update : {DM, GM, UM, AM, PM} →
{[DM], [GM], UM, [AM]}
generate : {DM, GM, UM, AM, PM} → {PM}

These adaptivity functions *a* can be written as (are equivalent to) IF-THEN rules or Condition-Action (CA) rules as defined in [26].

---

[15] which can be regarded also as a hierarchical inclusion relation.
[16] e.g., into weighted 'OR' relations, not further detailed here.

Automatic transformation from the domain model to the adaptation model means to interpret the existing DM to generate adaptation rules. This can be done at the adaptivity function level that is described above, or at a higher level of adaptation language or adaptive strategies (these levels are defined in [10]). That would mean that, instead of assigning a specific transformation for a given link type (or concept type), the same link (or concept) could be transformed differently, according to a different (e.g., pedagogically rooted) adaptation strategy. Here we are going to refer to low-level automatic transformations (CA level) and some adaptation language-level oriented automatic transformations.

Please note that normally the AM is supposed to work only with the data in the GM, as this is already pre-selected for presentation.

3.3.1. DM→AM: by Concept Attribute Type. Attribute types can be used to show only specific attributes in specific conditions. These conditions can be automatically deduced by the system (as in adaptivity) or triggered by the AHS user (adaptability).

Example 7: For instance, a specific automatic adaptive rule can express the fact that we only want to show the 'text' attribute of concept c1 after the 'title' and 'introduction' were read:

IF(c1.title.access='true' AND
c1.introduction.access='true')
THEN c1. text.available='true';

Please note that we wrote the condition for simplification purposes in this form, but that attribute states such as 'access' and 'available' are part of the user model[17]. In order for this to be a general automatic transformation rule, for any concept C in the DM, all concepts in the UM that reflect the DM should have also attribute states 'access' and 'available', and the following low-level rule has to be added to the AM:

IF(c.title.access='true' AND c.introduction.access='true')
THEN c. text.available='true';

If generic rules as the one above are permitted, for each such transformation [26] only one rule will be added. The number of possible rules to generate is potentially infinite, because it is dependent also on newly added states into the UM, which can be numerous. If we consider the case where only *s=2* such states can be added, as in the above example, and even more, we enforce the restriction that the 'access' state can only be found on the left side, while the 'available' state can appear on both sides of the rue, we obtain for the *flexibility degree*:

$$flex(1) = \left(\sum_{i=1}^{A_{min}} C(A_{min}, i)\right)^3 = \left(\sum_{i=1}^{A_{min}} \frac{A_{min}!}{(A_{min}-i)!\,i!}\right)^3$$

For $A_{min}=7$, $flex(1)=(87)^3= 658503$, which is a huge number. We obtain such a huge number because the events of having 'access' states on the left, 'available' states on the left and 'available' states on the right are independent, meaning that for each state determining the attributes that appear as 'access'-ed on the left side of the IF all combinations of attributes with 'available' on the left are possible, etc. So, basically, even for a very limited situation with 2 states and only generic rule generation, a great number of adaptive rules can be automatically written, based on the authoring goal (inferred or not by the system).

3.4. From Goal and Constraints - to Adaptation Model (GM→AM)

This represents automatic (adaptive, adaptable) AM generation from the GM, according to an adaptation strategy or technique (e.g., based on a pedagogical strategy or technique). This type of transformation is more natural to the design of the LAOS structure, as the existence of the GM model supposes a pre-selection of the material that is to be presented to the hypermedia user, according to some (pedagogical) *goal* and delimited by some (spatial, time, pedagogical, etc.) *constraints*.

3.4.1. GM→AM: by Link Type. The GM, as said, contains pre-ordered and pre-selected information from the DM. This structure can already be interpreted in terms of the adaptation that is to be performed on it. For instance, the GM allows 'AND' relations between concepts, as well as 'OR' relations with some weights.

Example 11: These can be used to express that all concepts in a 'AND' relation should be read, for instance:

IF ((c.name.access='true' OR c.contents.access='true')
AND link(c,c2,'AND',*))
THEN { c2.name.accessible='true';
c2.contents.accessible='true';}

Example 12: In a similar way, an 'OR' relationship can be interpreted as inhibiting the reading of the other concepts in the same relationship[18]*:*

IF ((c.name.access='true' OR c.contents.access ='true')
AND link(c,c2,'OR',*))
THEN { c2.name.accessible='no';
c2.contents.accessible='no';}

Example 13: A more informed version of the above would check the weight of the current concept, to see if it is

[17] more precisely, part of the overlay part of the UM, because the UM can contain also other attributes such as user's prior knowledge, user's interest, etc., that are not an overlay model of the DM (or GM).

[18] In such a case, an 'OR' relationship acts actually as a 'XOR'.

above some threshold, before deciding to inhibit another concept:

IF ((c.name.access='true' OR c.contents.access ='true')
AND link(c,c2,'OR',w) AND w>threshold)
THEN { c2.name.accessible='no';
c2.contents.accessible='no';}

In such a way, various constructs can be automatically added to the generic adaptation rules, directly by interpreting the *goal and constraints model*.

3.5. From User - to Adaptation Model (UM→AM)

The user model can be a simple overlay model of the DM (as in [26]) or a more extended model, represented also as a concept map, as defined in section 2.3. For the first case, the user model just generates variable-value pairs, which can enter conditions in adaptive rules or which can be modified by these rules. For the second case, not only the variable-values are important and interesting, but also the relationships between the concepts themselves, which together form the UM.

3.5.1. UM→AM: by Concept Attribute Type. In the *Example 14: To illustrate a pure usage of UM elements only to generate an AM rule, we consider the same state of 'interest' about a concept, which is extracted from the overlay model of the UM. We want a rule that displays everything in the concept, if this concept is of interest to the user. The conditions on the left side of the rule will be part of the HM, while the resulting action on the right side will be a part of the FM:*

IF (c.interest > threshold)
THEN { c.name.available='true';
c.contents.available='true';}

Please note that we have used for both sides concepts from the GMw (and not the DM).

Moreover, please notice that this rule is again a generic rule, which can be applied on all concepts in a concept map, therefore drastically reducing the workload.

3.5.2. UM→AM: by Link Type. Link type can only be used when the UM is itself a concept map. In this way, we can express for instance the fact that two states in the UM are related.

Here, however, we try to look at a different type of link between UM concepts. For this, let's consider the link of type 'influence'.

Example 15: We will add a rule saying that the interest in a subject c might decrease if the user is interested in another subject c2.

IF LINK(c,c2,'influence',*)
THEN { c.interest= c.interest – c2.interest;}

Example 16: Or if we want, for instance, to prevent infinite loops, we limit the application of this rule by adding an extra condition that the interest to be changed should at least be positive:

IF (c.interest > 0 AND LINK(c,c2,'influence',*))
THEN { c.interest= c.interest – c2.interest;}.

4. Discussion and Conclusion

Reducing the authoring burden has been identified as one of the major priorities in adaptive hypermedia [2] towards creating material for the semantic web. There are many ways of achieving this. In this paper we have approached the issue of improving and making AHS authoring easier by enumerating a number of different types of automatic (adaptive, adaptable) transformations that can be directly performed by the adaptive hypermedia authoring system, as shown in section 3. These possible automatic authoring techniques (or transformations) are based on the data design given by the LAOS model, which allows a concept-oriented approach for data design, analysis and usage. For exemplifying the transformations, we first reviewed LAOS, the five level AHS authoring model that allows 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).

Here we have shown a glimpse on the great number of different design possibilities that these automatic functions still allow, given the existing structure, showing that the authoring capacity is not inhibited by the added automatic authoring functionality. The range of possibilities of outcomes was computed in the form of a *flexibility degree*, which shows also the range of the adaptivity of the final system. We have introduced and computed the flexibility degree offered by such transformations for different example cases, and we have discussed the significance and extension possibilities of some of these transformations. Although these transformations have been discussed and analyzed separately (for instance, DM to AM transformation was analyzed apart from GM to AM transformation, etc.), in practice it is reasonable to expect that these transformations can be in parallel. The combination of different 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 a higher flexibility degree. We have not extended all the examples or computed the flexibility degree for all the cases, as the space in the paper did not permit it. Moreover, we have skipped some transformations, such as the ones from the GM to the PM. Instead, we have tried to give an overview of the flavor of the different possible automatic transformations, their applicability and their diversity.

It is interesting to consider, for future research, the combination of these automatic transformations and, e.g., presentation strategies bound to specific cognitive styles. We expect that applying such strategies would affect several layers at once. Another direction to pursue is to compare our work with and use specifications given by [15]; in [15], formal concept analysis is presented, that allows discovering of patterns between application data, on one hand, and the usage of concepts, relations and the semantics given by their hierarchies, on the other hand.

5. Acknowledgements

This research is linked to the EC project ADAPT (101144-CP-1-2002-NL-MINERVA-MPP).

6. References

[1] 2L690: Hypermedia Structures and Systems, Lecturer: Prof. De Bra. http://wwwis.win.tue.nl/~debra/2L690/

[2] Brusilovsky, P. Developing adaptive educational hypermedia systems: From design models to authoring tools. In: T. Murray, S. Blessing and S. Ainsworth (eds.): Authoring Tools for Advanced Technology Learning Environment. Dordrecht: Kluwer Academic Publishers, 2003.

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

[4] 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.

[5] 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.

[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., Evaluating Adaptive Hypermedia Authoring while Teaching Adaptive Systems. Proc. of SAC, Track ELS'04, ACM, 929-934.

[8] Cristea, A.I., Automatic Authoring in the LAOS AHS Authoring Model, Hypertext 2003, Workshop on Adaptive Hypermedia and Adaptive Web-based Systems.

[9] Cristea, A.I., Automatic Authoring in the LAOS AHS Authoring Model, Hypertext 2003, Workshop on Adaptive Hypermedia and Adaptive Web-Based Systems.

[10] Cristea, A.I., and Calvi, L. The three Layers of Adaptation Granularity. UM'03. Springer.

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

[12] Cristea, A., De Mooij, A. Evaluation of MOT, an AHS Authoring Tool: URD Checklist and a special evaluation class, CATE'03 (International Conference on Computers and Advanced Technology in Education) Rhodos, Greece, IASTED, ACTA Press, ISBN 0-88986-361-X, pp. 241-246.

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

[14] 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.

[15] Ganter, B., Wille, R. Formal Concept Analysis. Mathematical Foundations, 1999, Springer, ISBN: 3-540-62771-5.

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

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

[18] 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.

[19] LOM standard, http://ltsc.ieee.org/wg12/

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

[21] MOT online, http://e-learning.dsp.pub.ro/mot/ ; http://e-learning.dsp.pub.ro/motadapt/

[22] SCORM standard, http://www.adlnet.org/index.cfm?fuseaction=scormabt

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

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

[25] 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.

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