The three Layers of Adaptation Granularity

Alexandra Cristea[1] and Licia Calvi[2][1]

[1] 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
http://wwwis.win.tue.nl/~alex/
[2] Dept. of Eng. and Systems Science, University of Pavia
Via Ferrata 1, 27100 Pavia, Italy
+39-0382-505370
licia.calvi@unipr.it

Abstract. In spite of the interest in AHS, there are not as many applications as could be expected. We have previously pinpointed the problem to rely on the difficulty of AHS authoring. Adaptive features that have been successfully introduced and implemented until now are often too fine grained, and an author easily looses the overview. This paper introduces a three-layer model and classification method for adaptive techniques: *direct adaptation rules*, *adaptation language* and *adaptation strategies*. The benefits of this model are twofold: on one hand, the granulation level of authoring of adaptive hypermedia can be precisely established, and authors therefore can work at the most suitable level for them. On the other hand, this is a step towards standardization of adaptive techniques, especially by grouping them into a higher-level adaptation language or strategies. In this way, not only *adaptive hypermedia authoring*, but also *adaptive techniques exchange between adaptive applications* can be enabled.

1 Introduction

The ever-growing interest in AHS research [1,2,3,4,6,7,9,11,12,23,24,25] is not always counterbalanced by a deep investigation into its foundations and principles to go beyond both existing applications and accepted methodologies. What is felt as being missing is, on the one hand, more writing possibilities and facilities for authors and, on the other hand, the recognition of established standards to perform adaptivity. This paper aims at contributing to this basic discussion by introducing the idea of the granularity in adaptivity treatment and by modeling such granularity in a three-layer model.

The paper is structured as follows: in Section 2, we will better motivate the reason behind adaptivity granularity. In Section 3, we will present these three levels and discuss them in some details. In Section 4, we will discuss them in a current application, the system MOT, developed at the Eindhoven University of Technology.

---

[1] Licia Calvi is also affiliated to the University of Parma, Dept. of Italian.

2 Motivation

Previously [6] we have already noticed, in concordance with other authors [1], that the transition from linear, book-like hypermedia authoring is not easy. Authors are confronted with the heavy task of designing contents alternatives, adaptation techniques and ultimately, the whole user-interaction mechanism. It becomes unrealistic therefore to assume that they would not need support in this process. We already deduced that for adaptive courseware the authoring tool would have to offer tunable complexity and automatically perform many of the authoring tasks. The solution we offered [6] was to divide the authoring process into a layered model, grouped (for educational purposes) into *conceptual*, *lesson* and *adaptation and presentation* layer and to design the respective help and feedback for each layer, concentrating on automatizing tasks which are repetitive from one author to another. Here we concentrate on one specific layer, the *adaptation* layer. In this paper, we propose a three-layer model and classification method for adaptive techniques and populate the different levels: *direct adaptation rules*, *adaptation language* and *adaptation strategies*. This model is aimed at standardizing adaptation techniques at the different levels and therefore works towards exchanging adaptive techniques between different applications, as well as helps the authors of adaptive hypermedia by giving them higher-level handlers of low-level adaptation techniques. Authors therefore could only specify adaptation at the level of adaptation strategies and let the system "handle the details", i.e., fill-in the adaptation language and respective adaptation rules. Note that for adaptive hypermedia this idea of separation of the adaptation authoring into different design steps is not new. In AHAM, already a distinction is made between the initialization of the user model (IU), updating the user model (UU), and the generation of the adaptation (GA) [25] at the execution phase of the adaptation model. However, this division, although necessary, comes at a too late stage and is not expressive enough for the authoring process.

3 The three Levels of Adaptation Model

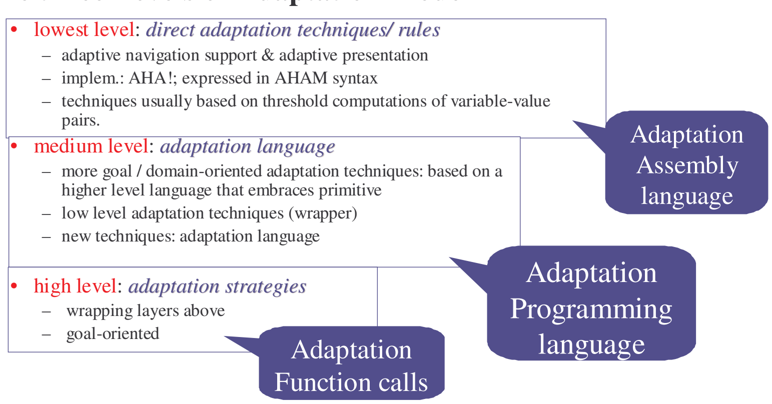

Fig. 1. The three layers of adaptation

3.1 Low level adaptation: *direct adaptation techniques*

Low-level adaptation techniques are all types of techniques previously used in adaptive hypermedia applications (*content adaptation* – adaptive presentation: inserting/removing of fragments, altering fragments, stretchtext, sorting fragments, dimming fragments - and *link adaptation* techniques: adaptive guidance – adaptive navigation support: direct guidance, link sorting, link hiding/ removal/ disabling, link annotation, link generation, map adaptation), summarized in [1,25].

They are usually determined by a mixture of fine-grained elements of the domain model (DM), user model (UM), adaptation model (AM), optionally (instantiated) goal model (GM) [7] and optional presentation model (PM). The adaptation engine (AE) works on these models, represented by all sort of types of links and concepts, variables or attributes[2] and their values, etc.

Adaptation at this level means defining a function *a*:

$$a : \{\mathbf{DM}, \mathbf{UM}, \mathbf{AM}, \mathbf{PM}\} \rightarrow \{[\mathbf{DM}], \mathbf{UM}, [\mathbf{AM}], \mathbf{PM}\} \quad (1)$$

Function *a* can furthermore be divided into a set of sub-functions:

$$a = \{update, generate\} \quad (2)$$

where:

$$update : \{\mathbf{DM}, \mathbf{UM}, \mathbf{AM}, \mathbf{PM}\} \rightarrow \{[\mathbf{DM}], \mathbf{UM}, [\mathbf{AM}]\}$$
$$generate : \{\mathbf{DM}, \mathbf{UM}, \mathbf{AM}, \mathbf{PM}\} \rightarrow \{\mathbf{PM}\} \quad (3)$$

Note that by defining *a* as a function, we already include the assumption that for each instance of the input values set $\{DM_i, UM_i, AM_i, PM_i\}$ the output $\{[DM_o]$[3], $UM_o, [AM_o], PM_o\}$ is uniquely determined, and thus *confluence* exists [23].

All these adaptivity functions *a* can be written as (are equivalent to) IF-THEN rules or Condition-Action rules as defined in [23].

3.2 Medium level adaptation: *adaptation language*

This level is determined by grouping the above into typical adaptation mechanisms and constructs (rules into higher-level adaptation rules; operators or language constructs and variables into adaptation language interface variables). The result is a 'programming language' for adaptation strategies [3], as listed in the following.

Most adaptive systems are rule-based and adaptation is mainly triggered by conditional rules. The mother of all rules is indeed a:

```
IF <PREREQUISITE> THEN <ACTION>                    (4)
```

rule. We can however elaborate on this rule and introduce a number of modifications that give rise to a set of additional adaptive rules. In [3] we presented a preliminary

[2] In the case of attributes, volatile and non-volatile [25] attributes are treated together.

[3] Normally, the AM and DM instances do not change as a result of applying *a*.

set of such conditional rules[4] and we showed their implementation in the AHA! system. One important derived rule is, for instance, a *level rule* [8], expressed as:

```
IF ENOUGH (<PREREQUISITES>) THEN <ACTION>        (5)
```

where ENOUGH = function of number and quality of prerequisites; true when, e.g., a given number of prerequisites from a PREREQUISITES set is fulfilled.

This type of relaxation of the prerequisites is intuitive[5], in the sense that it allows the author to write simplified rules, instead of writing a great number of complex ones.

Other rules that we have defined in [3] and that belong to the medium level of adaptivity, i.e., the adaptation language, are:

- A *temporal* rule: to capture unbound minimization, we need to add the WHILE construct in the original rule.

```
WHILE <CONDITION> DO <ACTION>        (6)
```

- A *repetition* rule: to indicate for how long a certain operation has to last before the reader can move on to another one.

```
FOR <i=1…n> DO <ACTION>        (7)
```

- An *interruption* command: to stop the user's current action.

```
BREAK <ACTION>        (8)
```

- A *generalization* command: to show the user a more general concept compared to the one s/he is currently reading.

```
GENERALIZE (COND, COND_1, …, COND_n)        (9)
```

- A *specialization* command: to show the user a more specific concept compared to the one s/he is currently reading.

```
SPECIALIZE (COND, COND_1, …, COND_n)        (10)
```

All these other possible rules (e.g., generalization, temporal, repetition, etc.) that can be developed from the original IF-THEN rule can be considered as deriving from Goldstein's Genetic Graph [14], because they, as well as Goldstein' graph, model the evolution of the user's knowledge during knowledge acquisition (e.g., abstraction, exemplification, generalization, etc.) and prescribe accordingly several ways in which the information nodes can be connected. In [3], we essentially concentrated on literary examples of this kind of practice simply to show, in a domain that follows rather peculiar strategies and design guidelines, how adaptivity may be considered in more generic terms and therefore applied to many more domains than the "simple" educational domain, which represents the most common exemplification of the adaptation philosophy. Such conditional rules allow more freedom, both in authoring as well as in navigating in this type of environment. They provide authors with more tools to express the kind on knowledge relationships they want to represent depending both on the inherent meaning they intend to deliver as well as on its context. At the same

---

[4] Based, among others, on [14,19,20].

[5] The idea is derived from game levels.

time, these rules are modeled according to the user's cognitive style and strategy in a way that we will make more explicit in the next section, where we will present some examples of the possible match between some adaptive strategies with possible cognitive styles.

3.3 Highest level of adaptation: *adaptation strategies*

There are several ways of modeling user's information processing strategies and cognitive styles. In [7], an overview of some of these models was discussed. Most of them rely however on the different ways in which people perceive and process information. In this sense, the four-parameter matrix we can derive looks like depicted in Fig. 2.

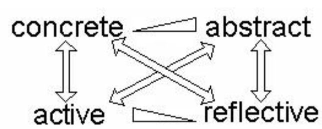

Fig. 2. The four-paradigm matrix for cognitive style modeling.

Here, the *concrete* users are those who need to comprehend information by doing, the *abstract* users are those who instead need more analytic insight, *active* users are those who need to see in practical use the new information they just acquired and the *reflective* ones are those who need to process and "metabolize" information before they can use it in practice.

To define how this model applies to all possible content domains, we would first need to identify for each such domain the specific function it satisfies. We can refer to this end to Jakobson's model [15], and extrapolate, from the functions he identifies, the three functions most information domains can be ascribed to: to *persuade*, to *inform*, to *instruct*. In an *instructional domain*, for example, we can assume that the adaptive strategies should be modeled so that:

1. Concrete students would get many practical drills and examples to work out.
2. Abstract students would be applied to an adaptation strategy that relies on generalization and specification rules.
3. Active students would get an adaptation strategy relying on exemplification.
4. Reflective students would be applied to an adaptation strategy that relies on generalization and specification rules and some examples.

These learning strategies can be complemented by some cognitive styles and some learning strategies considerations. So, in the first case, the adaptation strategies could be adjusted depending on the verbal or visual style (something that in the adaptive hypermedia literature is referred to as *preferences*) and on the either heuristic or systematic procedure the student chooses to process information. Verbal and visual learning modalities and heuristic and systematic learning strategies can also be orthogonally combined to fit all the four learning styles mentioned above.

So, for instance, a heuristic learning approach certainly fits the *converger* student profile [7]. For this profile, the adaptation strategy should therefore foresee a logically structured content, a series of learning tasks oriented towards learning to do, and self-assessment questionnaires. The *diverger* profile [7] would most probably feel more at ease with a systematic learning approach. Also in this case, learning tasks would mainly support learning-to-do. This will result in a mainly non-propositional [16] level of representation, where emphasis lies in the activation of the right sequence of operations to perform the task at hand. The *assimilator* profile [7] would be associated with learning-to-recall tasks due to its mainly theoretical orientation. Because of the emphasis on the construction of meaning, this kind of profile is linked to a mainly semantic, propositional level of representation [17,18]. In this case, the adaptation strategy would rely more on the use of generalization/exemplification rules to foster the student's deductive abilities and his/her attitude towards the creation of semantic associations between concepts. Finally, the *accommodator* profile [7] would be associated with a learning-to-do task and a trial and error learning strategy.

3.4 Examples

In the following part, we will show some schematic practical examples[6] of our model.

Table 1. Adaptive strategy for cognitive style: converger (abstract, active)

medium_increase() : *generate adaptive presentation with (obviously) increasing difficulty*
1. Explanation: Convergers are *abstract* and *active*; they like to feel in control; start with course for intermediates at medium adaptivity level, repeat for a number of times: - evaluate state of learner and start increasing difficulty & decreasing adaptivity level if result=good - evaluate state of learner and start decreasing level if result=bad
2. Translation at medium level: (ENOUGH shows here that the result is above an average result) AdaptLevel= 5; N=AskUser(); # this is to let user feel and be in control; levels: (1=min to 10=max) FOR <I=1..N> DO { SPECIALIZE (ENOUGH(Result)); IF (AdaptLevel>1) AdaptLevel--; GENERALIZE (NOT(ENOUGH(Result))); IF (AdaptLevel<5) AdaptLevel++; } # Note that adaptation level is not allowed to increase too much

3. Translation at low level: (the average can be implemented but takes more space)
DiffLevel = 3; AdaptLevel= 5; # note that here there is no predefined number of repetitions
IF <ACTION> THEN # Note that above we don't need the action of the user for triggering;
{ IF (Result1 +Result2)/2>5 AND DiffLevel<10 THEN # Note that 'enough' and specialize
{ DiffLevel++; IF (AdaptLevel>1) AdaptLevel--;} # must be redefined each time
IF (Result1 +Result2)/2<5 AND DiffLevel>1 THEN {DiffLevel--; IF (AdaptLevel<5) AdaptLevel++;}
}

Table 2. Adaptive strategy for cognitive style: diverger (concrete, reflective)

low() : *generate adaptive presentation with adaptively increasing difficulty*
Explanation: start with course for beginners at high level of adaptation, from general issues + examples, down + rest as in Table 1

[6] We keep the examples on purpose simple, enough just to compare at a glance the three levels.

2. Translation at medium level: (ENOUGH same as in Table 1) AdaptLevel= 10; GENERALIZE(); WHILE (not_finished) DO { SPECIALIZE (ENOUGH(Result)); IF (AdaptLevel>5) AdaptLevel--; # Note that we keep adaptation GENERALIZE (NOT(ENOUGH(Result))); IF (AdaptLevel<10) AdaptLevel++; } # level high here

3. Translation at low level:
DiffLevel = 1; AdaptLevel= 10;
IF <ACTION> THEN
{ IF (Result1 +Result2)/2>5 AND DiffLevel<10 THEN { DiffLevel++; IF (AdaptLevel>5) AdaptLevel--;}
IF (Result1 +Result2)/2<5 AND DiffLevel>1 THEN {DiffLevel--; IF (AdaptLevel<10) AdaptLevel++;}}

Table 3. Adaptive strategy for cognitive style: assimilator (abstract, reflective)

high() : *generate adaptive presentation with high difficulty and little adaptivity*
1. Explanation: start with course for intermediates at high level adaptation + similar Table 1
2. Translation at medium level: (ENOUGH same as in Table 1) SPECIALIZE(); AdaptLevel= 1; WHILE (not_finished) DO { GENERALIZE(ENOUGH(Result)); SPECIALIZE (NOT(ENOUGH(Result))); }

3. Translation at low level:
DiffLevel = 10; AdaptLevel= 1;
IF <ACTION> THEN
{ IF (Result1 +Result2)/2>5 AND DiffLevel<10 THEN DiffLevel++;
IF (Result1 +Result2)/2<5 AND DiffLevel>1 THEN DiffLevel--; }

Table 4. Adaptive strategy for cognitive style: accommodator (concrete, active)

medium_decrease() : *generate adaptive presentation with (obviously) decreasing difficulty*
- 1. Explanation: Accomodators like to feel in control; they want first examples and then theory.
2. Translation at medium level: (ENOUGH same as in Table 1) AdaptLevel= 5; N=AskUser(); # this is to let user feel and be in control; FOR <I=1..N> DO { SPECIALIZE (ENOUGH(Result)); IF (AdaptLevel>1) AdaptLevel--; GENERALIZE (NOT(ENOUGH(Result))); IF (AdaptLevel<5) AdaptLevel++; }

3. Translation at low level: (the average can be implemented but takes more space)
DiffLevel = 8; AdaptLevel= 5;
IF <ACTION> THEN
{ IF (Result1 +Result2)/2>5 AND DiffLevel<10 THEN {DiffLevel++; IF (AdaptLevel>1) AdaptLevel--;}
IF (Result1 +Result2)/2<5 AND DiffLevel>1 THEN {DiffLevel--; IF (AdaptLevel<5) AdaptLevel++;}}

4 MOT and the three Layers of Adaptation

In the following, we present the integration of the three layers of adaptation in an implementation example. We point at how these layers are reflected in the MOT system, an adaptive hypermedia authoring system developed at the Technical University of Eindhoven, and give some details of functionality for each layer.

4.1 Low level adaptation

As it can be seen from the description in section 3 and in [10,10], generation of low level adaptation techniques means defining the constituting elements of DM, UM, AM (or *a*), PM. MOT is implementing each of these models separately, as described in [9].

The DM is actually divided into two layers, a *domain* (or concept) layer and a *goal* (in particular, for educational applications, *lesson*) layer, as defined in our previous research [6]. These two layers are implemented with the help of concept maps. Concept maps are tuples $<\mathtt{C},\ \mathtt{L}>$, where $\mathtt{C}$ represents the set of concepts and $\mathtt{L}$ the set of links ($CM \subseteq \mathbf{CM}$, the set of all concept maps of the AHS).

Concepts in the domain layer are tuples $<\mathbb{A}_c, C_c>$ where $\mathbb{A}_c$ ($\mathbb{A}_c \neq \emptyset$) is a set of attributes and C_c a set of sub-concepts; concepts in the goal layer (*GM*) are defined by the tuple $<\mathbb{A}_g, C_g>$ where $\mathbb{A}_g$ (card($\mathbb{A}_{min}$)=2)[7] is a set of attributes and C_g a set of sub-concepts. The detailed definitions are contained in another paper [9]. An example of domain and goal layer concepts can be seen in Figure 3. As can bee seen in Figure 3 (right side), the goal layer already implements AND-OR connections and allows weights for the different concepts. It is the role of the AM to give the interpretation of these weights and AND-OR connections, based on the existing goal layer and the UM. In this way, for the same domain and goal model, different presentations can be generated by the AE. For MOT, we plan to connect the system to different adaptive presentation engines, including AHA! [11]. In this way, we can let the PM be generated by the AHA! AE, AM and UM, all working at the low level of adaptivity.

4.2 Medium level adaptation

MOT doesn't allow direct programming of adaptive rules at the medium level of adaptation, mostly due to the fact that the adaptive layer in MOT is not yet developed. We are planning to explicitly introduce such constructs, to test the possible enhancement of authoring ease. This level is for authors who want to design new strategies.

4.3 Highest level of adaptation: adaptation strategies

As said, MOT allows a goal (or lesson) level, where the adaptation strategies can be implemented. Each "lesson" at this level is goal-oriented and can represent a specific adaptation strategy. At the level of the adaptation strategy, the AND-OR connections in Fig. 3 have a meaning (*semantics*).

For instance, the selection in Fig. 3 right side is the basis of adaptation for *diverger* or *assimilator* users. It is a no-frills text-based content-oriented selection. For *converger* or *accommodator* users, a selection based on examples would be more benefi-

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

cial. The level selections (generalization, specialization) such as described in tables 1-4 can be translated in MOT into attribute selections for the lesson level, which can already be done easily (and automatically) in MOT (Figure 3, right side). We are planning to introduce more refined versions of automatic high level adaptation strategy implementations (tables 1-4), to make the task of the adaptive hypermedia designers easy. At the moment, the connection with the adaptation model and user model is not yet made.

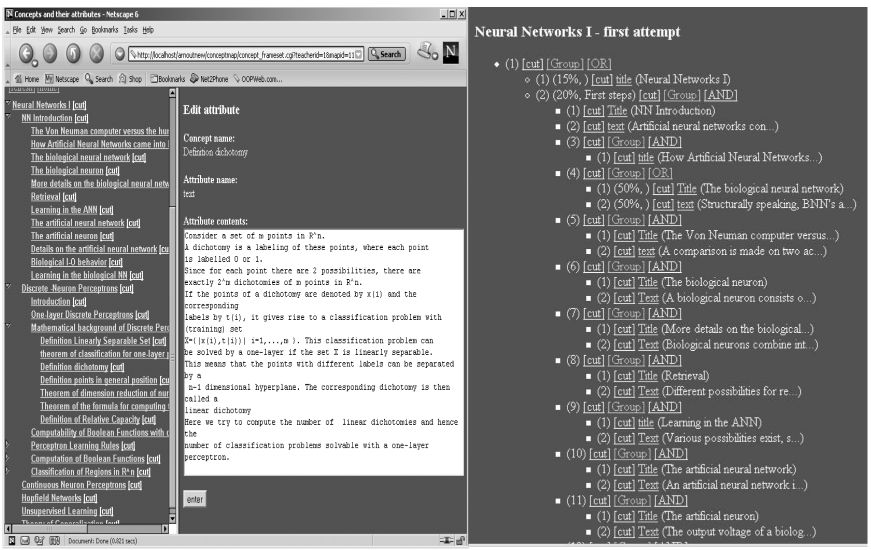

Fig. 3. Domain layer and goal layer concept editing in MOT

5 Conclusion

It is beneficial, especially for authoring purposes, to approach adaptation techniques from a higher level of semantics [3,25]. Next to the inherent difficulties of authoring adaptive hypermedia already highlighted in [3,6,25], the semantics of adaptation is difficult to follow at a low level of granularity of the implementation. In such cases, typical problems of CA can appear, such as no guarantee of termination and confluence [23,25]. These kinds of problems can be bypassed by allowing authors a gradual access to the adaptivity engine. For this purpose, we have defined three levels of adaptation: *direct adaptation techniques*, *adaptation language* and *adaptation strategies*. Moreover, these levels were instantiated with some examples based on MOT, a system implemented at the Eindhoven University of Technology. MOT is gradually implementing each of these layers.

The levels of adaptation that we have proposed therefore group adaptive techniques according to their *implementation form* (adaptation language) or *purpose* (adaptive strategies). In this way, we actually label the (groups of) adaptive techniques for further usage (re-implementation for authoring purposes or re-usage for exchange of adaptive techniques between adaptive applications). We are therefore working towards creating an ontology of adaptive techniques and integrating them in the new generation of meaningful Web [22], the semantic web [21].

References

1. Brusilovsky, P.: Adaptive hypermedia, User Modeling and User Adapted Interaction, Ten Year Anniversary Issue (Alfred Kobsa, ed.) 11 (1/2) (2002) 87-110
2. 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 WWW Conference (14-18 April 1998) 30 (1-7), 291-300
3. Calvi, L., and Cristea, A.I.: Towards Generic Adaptive Systems Analysis of a Case Study, in Proceedings of AH'02, LNCS 2347, Springer, (Malaga, Spain, May 2002) 79-89
4. Carro, R. M., Pulido, E. Rodríguez, P.: Designing Adaptive Web-based Courses with TANGOW .In proceedings of ICCE'99, V. 2 (Chiba, Japan, November 1999) 697-704
5. Cristea A.I., Okamoto, T.: Considering automatic educational validation of computerized educational systems, IEEE ICALT2001, Madison, USA, Okamoto T., Hartley R., Kinshuk & Klus J. (Eds.) (2001), CA (ISBN 0-7695-1013-2)
6. Cristea, A.I., and Aroyo, L.: Adaptive Authoring of Adaptive Educational Hypermedia, In Proceedings of AH 2002, LNCS 2347, Springer, 122-132
7. 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
8. Cristea, A., Okamoto, T.: MyEnglishTeacher – A WWW System for Academic English Teaching, in Proceedings of the ICCE 2000 Conference (Taipei, Taiwan, 2000)
9. Cristea, A., De Mooij, A.: Adaptive Course Authoring: MOT, My Online Teacher, in Proceedings of ICT-2003, IEEE LTTF, "Telecommunications + Education" Workshop (Feb 23 - March 1, 2003 Tahiti Island in Papetee - French Polynesia) (in press)
10. 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
11. De Bra, P. and Calvi, L.: AHA! An open Adaptive Hypermedia Architecture, The New Review of Hypermedia and Multimedia, V. 4, Taylor Graham Publishers (1998) 115-139
12. 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
13. Frasincar, F., Houben, G.J., Vdovjak, R., and Barna P.: RAL: An Algebra for Querying RDF, in Proceedings of WISE 2002 (Singapore, December 2002)
14. Goldstein, I.: The genetic graph: a representation for the evolution of procedural knowledge, In D. Sleeman and J.S. Brown (eds.), ITS, Academic Press, (1982)
15. Jakobson, R. (1960). Closing Statements: Linguistics and Poetics, in T. A. Sebeok (ed.), *Style in Language,* MIT Press, Cambridge.
16. Johnson-Laird, P.N.: Mental Models: Towards a Cognitive Science of Language, Inference, and Consciousness, Cambridge University Press, Cambridge, MA (1983)
17. Kintsch, W.: The role of knowledge in discourse comprehension: A construction-integration model, Psychological Review, 95 (1988) 163-182
18. Kintsch, W and van Dijk TA. Toward a Model of Text Comprehension and Production. Psychological Review, 85 (1978) 363-394
19. Lughi, G. Parole on line. Dall'ipertesto all'editoria multimediale. Guerini e Assoc., 2001.
20. Trigg, R.H. A network-based approach to text handling for the online scientific community. PhD thesis, Dept. of Computer Science, University of Maryland, 1983.
21. WC3, Semantic Web. http://www.w3.org/2001/sw/
22. W3C, Requirements for a Web Ontology Language, http://www.w3.org/TR/webont-req/
23. Wu, H., De Bra, P.: Sufficient Conditions for Well-Behaved Adaptive Hypermedia Systems, in Proceedings of WI'01, LNAI V.2198, Springer (Maebashi, October 2001) 148-152
24. 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
25. Wu, H.: A Reference Architecture for Adaptive Hypermedia Applications, doctoral thesis, Eindhoven University of Technology, The Netherlands, ISBN 90-386-0572-2