

# Yet Another Evaluation Framework

Carlos Miguel Tobar

PUC-Campinas, Computer Engineering Faculty, CP 317,  
13086-900 Campinas, SP, Brazil  
tobar@puc-campinas.edu.br

**Abstract.** Separation of concerns can play an important role in empirical evaluation of systems with adaptation characteristics, that of clarifying *what* to evaluate in order to prepare the *how* to evaluate. This present proposal integrates different design perspectives to better understand adaptation assessment and design through a comprehensive framework with: *abstract levels*, necessary to step-by-step approaches; services or *modeling issues* to support proper specification mechanisms for data manipulation; specification dimensions or *traditional-concerns*, defined for software modeling; and *goal conditions* behind the application descriptions, related to main external concepts. The result is a map to be used to perceive and exploit those different perspectives in evaluation as well as design of adaptation systems.

## 1 Introduction

*Models* guide representation production and are mainly used for data structure design in databases (DB) and artificial intelligence (AI), and for design and development in software engineering (SE). Models are mediating artifacts in the activity of design and provide designers with the ability to record their perceptions about a domain, which Benyon and Imaz [1] call *perspectives*.

Models usually support structural, behavioral, and constraint-related representations, but this is just a “traditional-concerns” perspective among several others.

Difficulties and complexities arise when software designers and implementers face systems with hypermedia or adaptation<sup>1</sup> requirements, and it is worst when both are necessary. Adaptation-Hypermedia (AH) systems are those with requirements of both hypermedia and adaptation in the Human-Computer Interaction (HCI) field. In particular, evaluation of AH systems has remained a challenge despite the growth of research efforts related to user interface adaptation [2].

Evaluation aims the comparison of “what something is” to “what it ought to be”, in order to facilitate a judgment about the value of that thing. Undoubtedly, in this pursuit, important SE activities such as verification and formal correctness have being used and studied under what is called software testing. Two basic and complementary points-of-view guide software testing: *black box* and *white box*.

Considering HCI, one of the black-box issues becomes very important and demands a separate treatment, within a new dimension, that of the “human factor”. Here appears

---

<sup>1</sup> *Adaptation* is a generalization that applies to both *adaptive* and *adaptable* mechanisms.

a more integrative sociological and psychological vision where the cognitive artifact, the AH system, should be holistically considered with the environment where it is used and with the external agents that interact with it (people and other systems).

Considering the human factor, HCI evaluation turns into *system acceptability* [3], which practically focuses satisfactory outcomes, mainly in the form of user's *satisfaction* and/or user's *task or activity realization* [4], [5]. Acceptability should be the final evaluation criterion for any HCI system, except when social processes are concerned, i.e., when the artifact is part of a broader social context.

Models are important for exploring, testing, recording and communicating designs [1], and essential for proper adaptation evaluation [5].

Hypermedia data models have been proposed in order to face hypermedia as a paradigm to develop information-oriented systems, especially those related with HCI. Hypermedia demands *ad-hoc* requirements not found in conventional DB, AI, or SE modeling, such as data navigation, data perception, and interaction [6], in addition to system evaluation.

Combinations of models have been exploited as development methodologies, e.g., UML<sup>2</sup> and OOHDM [7], because there are different and complementary perspectives that must be considered in order to develop complex systems.

When it comes to better design, develop, maintain, and assess AH systems, proposed frameworks and models, with almost no exception, present combinations of DB data entities or SE functional components, sometimes with abstractions layers. But, a more comprehensive framework is needed where abstractions levels, design issues, traditional-concerns, and goal conditions of applications are adequately separated but integrated, since they are all perspectives of a whole. The result is a proposal for a more structured vision for AH designs, which is consistent with SE test-principles.

This paper is organized to initially discuss some important work on frameworks, models, and reference models related to AH. Following, there is an evaluation-plan discussion that is based on one architectural model oriented to assessment. Those discussions are worth to support the proposal of a different integrative framework that can be used to orient AH assessment and design, once it is taken for granted that evaluation should be considered during all the development cycle of a software system.

## 2 Frameworks, Models, and Reference Models

In this Section, it is presented a revision on frameworks, models, and reference models that have been proposed to design and assess AH systems or that have strong influence on the topic, in order to highlight the existence of different modeling perspectives.

Four complementary perspectives should be considered in AH system design and assessment: *abstract levels* to step-by-step approaches for bottom-up or top-down development; services or *modeling issues* to support proper specification mechanisms for data manipulation; specification dimensions or *traditional-concerns*, defined for software modeling (e.g. the process concern); and *goal conditions* behind the application descriptions, related to the main external concepts that must be considered, i.e., matters that the application realization must be concerned with.

---

<sup>2</sup> UML™ (<http://www.uml.org/>) is trademark of OMG (<http://www.omg.org>).

Frameworks for information systems, and in especial for AH systems, are necessary to “much better interpret and give more exact hints for failures and false inferences than a simple global vision, thus facilitating the improvement of applications and services, when required, as well as the generalization and reuse of results” [4].

Architectures have been proposed and used as frameworks for models, defining modeling perspectives, such as the ANSI/SPARC architecture [8], important reference to abstraction levels to allow categorization of DB schemas; and the Dexter Reference Model [9] for hypermedia design that pioneered the separation of the navigation hypermedia issue from those of presentation and interaction.

De Bra and colleagues [10], based heavily on the Dexter Model, proposed a reference model that is an extension for AH applications. Their extension, AHAM, is related to the *storage* functional layer, with its division into three models: *domain*, *user*, and *teaching*, in addition to an adaptive engine. Adaptation of content and links is separated to each user and makes authoring easier, but is too restricted. Separation between presentation and interaction could also be considered. AHAM does not cover authoring, and mixes modeling issues with information components.

The use of taxonomies is another trend to separate categories and better understand complex matters. The most cited taxonomy in AH literature is due to Brusilovsky [11], which focuses on the interface and user interaction, and has been divided into two distinct areas: *adaptive presentation* and *adaptive navigation support*. Again, at least, a refinement is desirable to separate the presentation issue from the interaction one.

Some frameworks and models aiming a better understanding of adaptation are considered *adaptation oriented* and rely on different architectural dimensions for software systems: *data-centered oriented*, *process-centered oriented*, or *hybrid*. Data-centered proposals follows very close the AHAM reference model.

The main idea behind adaptation-oriented frameworks is to break down a monolithic AH system into several components, which can be understood and treated separately. But, among the considered proposals there are no concerns with abstractions levels, modeling issues, or goal conditions, and, where the traditional-concerns perspective is applicable, at least one of the specification dimensions is missing.

Benyon and Murray [12] introduced a data-centered architecture for AH systems that focuses on adaptation of information models. The architecture basically consists of: the *user model*, the *domain model*, and the *interaction model*. Triantafillou and colleagues [13] also used a data-centered vision for AH systems with similar models. In the learner model there are different categories of information: *personal profile* (static data), *cognitive profile* (cognitive style and controls), and an *overlay learner knowledge profile* (concepts).

Process-centered architectures were proposed by Karagiannidis and Sampson [4], as a layered evaluation framework with an *interaction assessment* layer for evaluation and an *adaptation decision making* layer; and by Weibelzahl [5], whom proposed a more detailed information processing model of adaptive systems for assessment purposes, with four main layers: *evaluation of reliability and external validity of input data acquisition*; *evaluation of the inference mechanism and accuracy of user properties*; *appropriateness of adaptation decisions*; and *quality of total interaction*, including in this last one *system behavior*, and *user behavior and usability*.

Paramythis and colleagues [14] proposed a hybrid architecture with five components, four of them process oriented: *interaction monitoring*, *interpretation /*

*inferences, adaptation decision-making, and applying adaptation*; in addition, there is *modeling*, a data centered component. There are also three other optional components: *explicitly provided knowledge* (data centered), *transparent models and adaptation rationale* (process oriented), and *automatic adaptation assessment* (process oriented). Adaptation evaluation is directed to combinations of these components.

Efforts to integrate perspectives were initially directed toward Open Hypermedia systems, which are roughly related to general hypermedia systems usually without adaptation concerns. For instance, an architecture that is continually evolving is OOHDM [7], which is a development methodology composed by five stages: *requirements gathering, conceptual design, navigational design, abstract interface design, and implementation*. OOHDM tackles SE issues, lacks a better separation of design issues, and needs to formally address goals and conditions.

Another integrative framework is the ACM – Abstract Categorization Map [6], presented as a graphical tool. Through ACM, it is possible to assess and compare existing specification mechanisms in data models oriented to hypermedia, regarding three different points-of-view: *abstraction level of the modeling; data manipulation services* that are considered; and *dimensions for data specification*. The map allows analysis of existing models as an exercise to relate them, and therefore to derive strengths, weaknesses, overlaps, or omissions. The ACM was defined without considering adaptation matters and has an extension proposed elsewhere below.

### 3 Developing an AH Evaluation Plan

As base for an evaluation plan, one of the architectures for assessment has to be chosen, preferably with all modeling perspectives that are reviewed above. Related issues and elements of this plan are discussed next.

There have been basically two architectural visions for AH assessment: the *global-system* and the *adaptation-oriented*. For the first one, assessment is based on traditional empirical research, usually with lots of quantitative or qualitative data, where humans are used as participants (subjects) in laboratory experiments. In the second one, some sort of separation of concerns is used to localize and fix problems, which are usually detected through previous assessment efforts conducted under the global-system vision.

The second vision is formative in essence, as can be observed in the goals of the reviewed frameworks, models, and reference models, e.g., a reference model aims to present common abstractions and to provide basis for development of a specific application type; or in the work by Benyon and Murray [12], which aims a methodology for development with evaluation as central activity.

Separation of concerns can play an important role in empirical evaluations, that of clarifying *what* to evaluate in order to prepare the *how* to evaluate. Answers to these questions constitute an evaluation plan that must be based on the chosen architecture for assessment, as well as on the *goals* for the AH system. Of course, organization policies, and restrictions for time and resources should also be considered.

Goals are what the user and/or the AH system have to accomplish and represent the *why* of the system existence. Goals determine *objects of assessment* that should be described through abstract very high-level models, in terms of the application context and as many conditions as necessary.

Once objects of assessment are clearly determined, it is time to define what features on them are worth to evaluate. Those features should reflect different, independent, and measurable *quality factors*, the so-called *criteria*. Some goal conditions can be expressed in terms of explicit quality factors such as *accessibility* or *user performance*.

Research regarding user-oriented assessment in HCI, because of its foundation on psychology, has always been tightly associated with usability, considered the main quality group of factors. There is almost an usability-sole view for assessment that has progressively been augmented with engineering issues, such as efficiency, error recovery, precision, and recall, among a very long list of vague, confusing, and fuzzy concepts, which are presented as assessment criteria, metrics, measures, methods, or techniques, in an intermixed way.

A clear categorization of quality factors offers a good opportunity to separate human factors from the ones related to engineering. The latter have received attention of SE research and can be assessed by SE principles. In the context of the former, out of the broad range of available evaluation techniques, only a few can be applied [2] and thus deserve a more detailed planning.

Nielsen's attribute taxonomy [3] for system acceptability can be considered as the ground for the definition on human-factor major categories (utility and usability). For the engineering major group-factors (functionality, reliability, efficiency, maintainability, and portability), the ISO9126 specification [15] can be considered, except for the quality factor of usability inside it, a human factor. All of these categories present sub-categories that help the planning effort.

In order to define the evaluation plan, is good to note that the set of possible quality factors is complementary optimal, in the sense that an AH system does not need to offer acceptable characteristics on all criteria. Some goal conditions can involve directly or indirectly quality factors, and even can establish non-applicable or semi-applicable grounds for one or more criteria, e.g., if one defines as condition the compliance of the AH system to a standard middleware platform that is ready, there is little or no need at all to assess interoperability matters, one of the functionality issues.

In the form of a question or case study, each feature to be evaluated in an object of assessment corresponds to an assessment dimension that has to have a viable and adequate *method of assessment*, which should allow the definition of a clear measure.

Assessment of HCI human factors has demanded new forms of evaluation methods, other than the traditional empirical-quantitative-testing methods that produce statements about a highly standardized, repeatable, precisely described, person-independent, averaged situations. The main reason for this demand is that these traditional evaluations have been conducted independently from the individual, the optimal target of adaptation. Paramythis and colleagues [14] present an interesting set of evaluation methods directed to the human-factor of usability, where it is possible to observe their application in formative as well as summative assessment.

Assessment of engineering-factors for AH systems has been conducted along the traditional SE black-box summative approach, although, it seems, not in conscious ways. It usually depends on previous assessment efforts that point to failures or misconceptions that are used as clues for the isolation and localization of errors.

Additional visions for AH assessment can be perceived when separation of concerns is combined with the white-box approach. In this case, an object of assessment is one of the architectural components of the AH system, which has a feature assessed:

- against some outcome resulted from the interaction between user and AH system, such as a *performance result* or the *user's behavior*; or
- according to known and desirable criteria, which allow automatic processing of quantitative data through established metrics, such as *fan-in* or *fan-out* in a hypermedia page.

The global-system vision can also gain if used together with the white-box approach, but, as pointed out by Brusilovsky and colleagues [16], detected problems can be hard to localize in order to be solved, and, even with a successful adaptation result, there can be *minus-times-minus-equal-positive* results.

#### 4 The Extended Abstract Categorization Map

The Extended Abstract Categorization Map (E-ACM) is proposed in order to guide adaptation evaluation and design. It also supports, as a guide, the choice of models.

An educational application is used as example to provide better understanding of the proposal. Basically, AH systems can assume three basic forms in the educational domain: *recommender* (including *advisors* and *helpers*), *content tailor*, and pedagogical *tutor*. This proposal is applicable to either of these systems as well as any other AH system, since the information of any type of AH system can be rendered through an instructional strategy. The chosen system is a content tailor for an undergraduate course in computer engineering, on distributed operational systems.

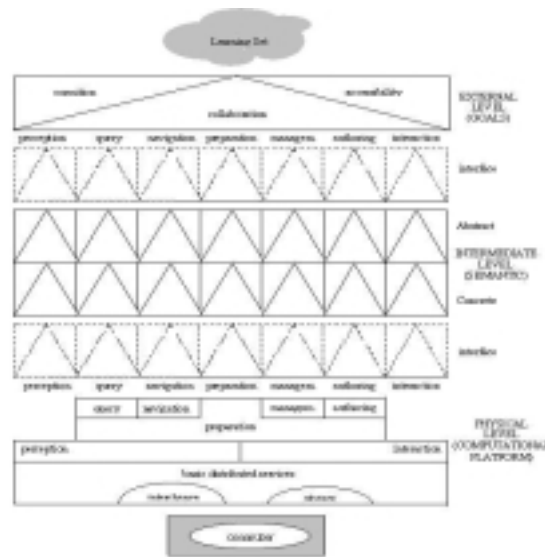


Fig. 1. The Extended Abstract Categorization Map

The E-ACM (Figure 1) covers a broad set of characteristics for AH systems, i.e., it aims to cover complementary perspectives for AH system modeling.

The E-ACM separates, a priori, two of the modeling perspectives: on one extreme, on the top, goal conditions for the application are considered, in this case three conditions; on the other extreme, on the bottom, data manipulation services are considered. Columns and cross polygons (triangles because of the three conditions) are created when conditions and services are projected toward each other through the intermediate abstraction level and abstraction interfaces. There are three abstraction levels: external, intermediate, and physical, which are separated by interfaces.

Each resulting column represents the influence of one service through all of the intermediate abstraction sublevels and interfaces. Each cross polygon represents the influence of one goal condition at one of the semantic abstraction sublevels or one interface, relative to the respective column service. This allows a separation of concerns approach to specification modeling and/or adaptation evaluation, which also comprehends complete traditional-concerns: data, process, and constraints.

Traditional-concerns change according to the application domain. In the case of an educational AH system, the data concern is usually related to three main data structures: learner, content, and learning design. The process concern is related to three main processes: input acquirement and inference, adaptation selection, and adaptation realization. And, last but not least, there are the applicable constraints. This means that from the intermediate abstract level, all through to the physical level, orthogonal to the map, there is a traditional SE design with all its components, on the ground of a hybrid adaptation-oriented framework with the addition of the constraint-related specification.

In the present context, a good start point is the external level, where, first of all, main goals should be defined. The main goal in the considered example is to improve learning outcomes through adaptation exploitation, in order to allow students the opportunity to know, design, and develop basic operational software with distribution characteristics. The three pre-defined conditions are: adaptation according to *cognitive* differences between students; task information should be found and worked *collaboratively*; and the learning environment should be *accessible* to any student.

Given goal and conditions, the object of assessment is defined to be the learning practice through the AH system. Considering only the cognition-related condition, because of space constraints, it is determined the use of a learning model that defines students' stereotypes through cognitive styles. Those styles point to a number of assessment dimensions directly dependent on the user's learning characteristics.

At this point, the evaluation plan is being defined by the determination of the assessment criteria. Under the criteria of *usability satisfaction*, *utility satisfaction* and *operability*, chosen among a dozen of human factors, an analysis upon existent learning outcomes and quality factors could point out to focus the evaluation on the following question: *is there any learning outcome improvement through adaptation exploitation?*

Considering goals, conditions, criteria, and assessment question, named *assessment guides* from here on, educational contents and tasks should be structured with the support of external models, such as concepts maps [10], [17], or UML use-case diagrams, which can be used to represent different teaching-learning scenarios.

Models on top of the map help individuals perceive the main goals in the problem domain. They are used to identify and represent information elements, their characteristics, and relationships, without any concern with low-level descriptions in a computational platform, such as realizations of navigational links or anchors that are software elements.

Following to the intermediate level, several different tasks could be considered in parallel. Some are design oriented and some are assessment oriented. Both types of tasks are mutually influenced.

In each of the map polygons, the use of intermediate models introduces abstract mechanisms related to the computational world, which facilitate application development. Ideally, an intermediate model should allow the transformation of information elements into computational ones, preferentially through mappings that begin at the external level, somewhat abstractly, into implementation specifications.

Mappings between the levels are required in order to progress from one level of description to another [1]. The easiness of a mapping between models that belong to different abstraction levels depends on available mechanisms supported by the higher-level model, named *mapping facilitators*, which offer some direct translation type for the elements specified through them to elements obtained through the lower-level model. For instance, a relationship represented with the relational model can be used to identify links in a navigational model.

The intermediate level is divided into two sublevels to allow the localization of models and modeling results regarding the influence of the other two levels. If the influence is due to the external level and its information goals, the modeling belongs to the abstract sublevel. If, otherwise, the influence is due the physical level and its data services, then it belongs to the concrete sublevel.

Continuing the example, at the abstract sublevel of the navigational service, it is possible to use object-oriented views, object-oriented state charts, and context classes to produce the abstract navigational design, such as is professed in OOHDM [7]. For other services, other models should likely be used.

In terms of assessment, during the design in a cross-polygon of E-ACM, considering the assessment guides, special inbuilt data-gathering mechanisms can be considered.

In parallel to designing, the evaluation plan is detailed by choosing proper methods and adequate measures for each question criterion, e.g., one can use questionnaires and observations to assess the level of user's satisfaction on the utility factor. Under SE orientation, engineering criteria should also be considered and observed.

At the concrete sub level, it is possible to abstract each foreseen adaptation through DB mechanisms, and thus do adaptation modeling through traditional DB techniques. Trigger-associated rules or scripts are responsible by the appropriate adaptation-effect selection, which, in turn, is associated to one or more computational services, with its timings (duration) and dynamics. Trigger conditions should be specified on data elements related to the data structures or to the computer platform. Triggers can also be modeled as condition rules a la AHA [10], level rule based on time elapsed [17], or overlay graph model with importance coefficients and weights [18].

At the bottom of the map are considered physical DB schemas, software and/or hardware information sensors, and programming code external to the databases in order to map the semantic results of the intermediate level. These elements responsible by data manipulation for the proper interaction between system and students are specified through different software utilities that support physical models.

Aside of empirical evidence, during the design process through the different abstractions levels, tools that support modeling can offer mechanisms that stress quality assurance, which can assure a robust design and a better final system, such as verification of completeness or reference closure. Mapping facilitators can also be used

as assessment elements for adequacy of transformations between abstraction levels.

The E-ACM also can be used in summative assessment, in the same direction pointed out by Paramythis and colleagues [14], i.e., not to treat adaptation as a monolithic process, rather, it should be broken down into its constituents, which should be evaluated separately where necessary and feasible. The map offers a greater number of perspectives to identify and localize errors, separated by manipulation services in addition to traditional-concerns components.

## 5 Conclusions

Separation of concerns, or the well-known SE divide-to-conquer strategy, is used to ease specification and programming, reducing costs of development, verification, validation, and maintenance. This central SE practice is proposed as the base of a new framework for AH systems, because it facilitates evaluation processes through separating design perspectives.

While hypermedia is more than navigation and presentation, spanning other data manipulation services, adaptation has been perceived through confused perspective combinations of data and process approaches. In order to orient and facilitate the choice of models to be used during AH system evaluation and development, a more comprehensive framework is proposed in the form of a map, the Extended Abstract Categorization Map, integrating abstract levels, data manipulation services, traditional concerns, and goal conditions of applications.

Human and engineering factors, related to the user, interaction process, and computational system, should guide evaluation in the form of an evaluation plan that is primarily based on goals and conditions, and whose realization have to happen together within each modeling activity.

The traditional software life cycle is behind the map proposal, but adaptive software evaluation is not just the last phase and is seen as an important source of information throughout the complete cycle [3]. Although the impression of a strait sequential effort, on the contrary, there is consciousness for the need of evaluation and re-design of AH systems, whose requirements usually cannot be fixed and used as the basis for deriving a formal design equivalent [19]. Usually, designers do not really know the problem until they start working on its solution, requiring some sort of iterative dialogue between themselves and the design [20].

In addition, to be better elaborated, an open framework for AH systems must allow the consideration of personal, social, and cultural interests, as well as variable and flexible evaluation methods.

## Acknowledgments

This work was supported in part by CNPq, a Brazilian Governmental Agency that fosters scientific and technological development.

Carlos Tobar, as visiting researcher, thanks Prof. Gerhard Fischer and all his staff within the L3D – Center for LifeLong Learning & Design, Department of Computer

Science and Institute of Cognitive Science, University of Colorado at Boulder.

The author also thanks the very valuable suggestions received from the reviewers.

## References

1. Benyon, D. and Imaz, M., "Metaphors and models: conceptual foundations of representations in interactive systems development", *Human-Computer Interaction (HCI)*, vol. 14, Issue ½, pp. 159-189, 1999.
2. Akousmianaki, D., Grammenos, D., Stephanidis, C., "User Interface Adaptation: Evaluation Perspectives", in *User Interfaces for All: Concepts, Methods, and Tools*, Stephanidis, C. (ed), 2001, Lawrence Erlbaum.
3. Nielsen, J., *Usability Engineering*, Morgan Kaufmann, 1994.
4. Karagiannidis, C. and Sampson, D., "Layered evaluation of adaptive applications and services", *Int'l Conf. on Adaptive Hypermedia and Adaptive Web-Based System*, AH2000, 2000.
5. Weibelzahl, S., *Evaluation of Adaptive Systems*, PHD Dissertation (to be) Presented to the Faculty I of the University of Trier, Pedagogical University Freiburg, 2003.
6. Tobar, C. and Ricarte, I., "Towards a categorization of hypermedia data models", in *Multimedia Modeling*, A. Karmouch (ed), Singapura: World Scientific, pp. 79-95, 1999.
7. Rossi, G., Schwabe, D., and Guimarães, R., "Designing personalized web applications", *10<sup>th</sup> Int'l Conf. on the WWW: WWW10*, pp. 275-284, 2001.
8. Tschritzis, D. and Klug, A., "The ANSI/X33/SPARC DBMS framework: Report of the Study Group on Data Base Management Systems", *Information Systems*, vol. 3, 1978.
9. Halasz, F. and Schwartz, M., "The Dexter hypertext reference model", *CACM*, 37(2), pp.30-39, 1994.
10. De Bra, P., Houben, G.-J., and Wu, H., "AHAM: A Dexter-based Reference Model of Adaptive Hypermedia", *10<sup>th</sup> ACM Hypertext and Hypermedia Conf.*, pp. 147-156. 1999.
11. Brusilovsky, P., "Methods and techniques of adaptive hypermedia", *Journal of User Modeling and User-Adaptive Interaction*, 6(2-3), pp. 87-129, 1996.
12. Benyon D. and Murray, D., "Adaptive systems; from intelligent tutoring to autonomous agents", *Knowledge-Based Systems*, 6(4), pp. 197-219, 1993.
13. Triantafillou, E., Pomportsis, A., and Georgiadou, E., "AEC-CS: Adaptive Educational System based on Cognitive Styles", in *Adaptive Systems for Web-based Education*, Proc. of the AH2002 Workshop on Adaptive Systems for Web-Based Education, pp. 10-20, 2002.
14. Paramythis, A., Totter, A., and Stephanidis, C., "A modular approach to the evaluation of adaptive user interfaces", in *Workshop on Empirical Evaluation of Adaptive Systems*, 8th Int'l Conf. UM, pp. 9-24, 2001.
15. International Standards Organisation, ISO / IEC 9126-1, *Software engineering -- Product quality -- Part 1: Quality model*, 2001.
16. Brusilovsky, P., Karagiannidis, C. and Sampson, D., "The Benefits of Layered Evaluation of Adaptive Applications and Services," in *Workshop on Empirical Evaluation of Adaptive Systems*, 8th Int'l Conf. UM, pp. 1-8, 2001.
17. Calvi, L. and Cristea, A., "Towards Generic Adaptive Systems: Analysis of a Case Study", in *Adaptive Hypermedia and Adaptive Web-Based Systems*, LNCS 2347, pp. 79. 2002.
18. Cristea, A. and Aroyo, L., "Adaptive Authoring of Adaptive Educational Hypermedia," in *Adaptive Hypermedia and Adaptive Web-Based Systems*, LNCS 2347, pp. 122, 2002.
19. Fischer, G., "HCI Software: Lessons learned, challenges ahead", *IEEE Software*, 6(1), pp. 44-52, 1989.
20. Fallman, D., "Design-Oriented Human-Computer Interaction", *Proceedings of CHI2003, Conference on Human Factors in Computing Systems*, pp. 225-232, April 2003.