Self-adaptation of educational software

Margherita Antona\textsuperscript{1}, Anthony Savidis\textsuperscript{1}, Alexandros Paramythis\textsuperscript{1}, Constantine Stephanidis\textsuperscript{1,2}

\textsuperscript{1}Institute of Computer Science
Foundation for Research and Technology - Hellas
Science and Technology Park of Crete
GR-71110, Heraklion, Crete, Greece
cs@ics.forth.gr

\textsuperscript{2}Department of Computer Science
University of Crete

Abstract. This paper focuses on the notion of content and interaction adaptation in educational software. The adopted perspective is that adaptation should continuously deliver, at any time during the use of the system, the most appropriate interactive learning content to each individual learner. To this purpose, the paper briefly discusses the concept of adaptation in educational applications from the wider perspective of universal access to Information Society Technologies, and discusses some aspects of a proposed architectural framework suitable for supporting content and interaction adaptation in such a context.

1 Introduction

Computer-based instruction, and in particular Web-based instruction, is progressively acquiring a fundamental role in shaping new inclusive forms of education in the emerging Information Society, and is attracting considerable attention at both policy and industrial level [see, e.g., Commission of the European Communities, 2000a; Commission of the European Communities, 2000b; Kerrey et al., 2000; Hodgins, 2000]. The advent of multimedia technologies and of the World Wide Web has marked a turning point in the way computer-based education is conceived, designed, delivered and experienced. The WWW enables the wide distribution of educational material in which content is presented using a wide variety of media (hypertext, graphics, animation, audio, video, and so on). Web-based learning involves an increasing variety of target learner groups, including, for example, adults, children in secondary education, people in geographically dispersed areas, (re-) trainees, disabled and homebound persons, etc., in an increasingly wide variety of educational activities, including pre-university and university courses delivered on the Web, training in industry, life-long learning, learning on demand, and many others. As
a consequence, distance learning is evolving from a form of education in which the learner and the teacher are physically separated, to the provision of “whatever educational opportunities to anybody, anywhere and at anytime, characterised by a great diversity in practice” [Spodick, 1995, p.1].

The role of technology, in this context, is not only limited to providing more efficient, simple and cost-effective solutions for overcoming distance and time in education, but also to create and make widely available new forms of learning, exploiting the capability of dynamic electronic content to be manipulated, structured, retrieved and presented in a variety of ways. In this respect, one of the most promising characteristics of Web-based learning is the concrete possibility of providing adaptation and individualisation of learning resources on the basis of learners’ characteristics such as, for example, learning attitudes and strategies, background knowledge, and purposes of learning.

In recent years, the concept of adaptation has been investigated in the broad perspective of providing built-in accessibility and high quality interaction in applications and services in the emerging Information Society [Stephanidis, 2000; Stephanidis, 2001]. In this perspective, adaptation characterises software products that automatically adapt themselves according to the individual attributes of users (e.g., mental / motor / sensory characteristics, preferences), and to the particular context of use (e.g., hardware and software platform, environment of use).

In the application domain of education, adaptation has been investigated in a number of recent research efforts, mainly falling into the category of Intelligent Tutoring Systems and Adaptive Hypermedia. The goal of Intelligent Tutoring Systems is to use the knowledge about domains, learners, and teaching strategies to support flexible individualized learning and tutoring [Brusilovsky, 1999; Murray, 1999]. Adaptive hypermedia systems apply different forms of user models to adapt the content and the structure of hypermedia pages to the user, and have found in education one of their main applications [Brusilovsky, 1996; De Bra, 1999].

This paper reports on a preliminary investigation of some of the most important issues related to interaction and content adaptation in Web-based courseware. The adopted perspective is that adaptation should aim to be learner-centred in the first place, i.e., it should respond to the educational and interaction needs, requirements and preferences of individual learners.

2 Adaptation of interaction and content in educational software

As previously mentioned, educational technologies are evolving, from tools conceived to enable knowledge delivery at distance, to integrated environments supporting a wide variety of interactive and cooperative activities towards a broad range of educational goals. A recent extensive overview of course management software [FutureU, 2000] identifies a number of activity types for which tools are available. Learner-oriented activities include educational material viewing, browsing, searching, annotating, and personalising, cooperative teamwork, self-assessment exercises, study-skill building, and many others. Collaboration activities amongst
learners and between learners and teachers are subcategorised into discussion (synchronous and asynchronous through a variety of channels) and sharing. Finally, teaching-oriented activities include authoring of educational material, course management including learners’ assessment, and administrative tasks, as well as more general features such as access to on-line information, library search, etc.

As a consequence, interactive environments for education should provide integrated support for a broad range of activities, which involve users in different roles (learners, teachers, facilitators, domain experts, authors of educational material, administrative staff, etc.), and the vast majority of which exhibit interaction- and knowledge-intensive properties. Interactivity and cooperation between teachers and learners, learners and content, learners and the learning environment, and among learners themselves, constitutes therefore a critical “connectivity” factor [Saba, 2000; Sherry, 1996], greatly affecting the design and implementation of educational technologies and environments.

Another important factor that needs to be considered is the increasing user diversity along multiple dimensions. Apart from obvious diversity in age, abilities, language, cultural background, familiarity with technology, geographical position, etc., learners vary, on an individual basis, in educational aims and goals, available time and budget, preferred modes of learning (e.g., type and amount of required teacher feedback and support, and communication and cooperation with other learners), and learning styles and strategies. The latter two are particularly important, since enabling learners to follow interactive learning processes that suit their individual cognitive styles is considered one of the main factors influencing success [Sherry, 1996]. This calls for a learner-centred approach to the design and implementation of educational software products [Bates, 1995; Graves, 2000; Spodick, 1995], wherein personalised forms of learning are supported by taking into account user requirements and preferences throughout the whole product lifecycle. Analogous considerations hold for teachers and facilitators, who need to feel comfortable with technology and free to adopt their individual teaching and communication methods and strategies.

Furthermore, interaction in educational courseware is affected by the nature of the different contents to be communicated and assimilated, and by the type of educational material (e.g., teaching and training courses, libraries, knowledge repositories, video archives) available through a variety of different media (e.g., real time video, Web, multimedia and virtual reality applications). Content, media and technology have to be carefully matched, so as to best exploit the possibilities offered by different media and techniques in learner-content interaction, and to avoid technology-driven approaches [Sherry, 1996].

Additional dimensions of diversity are represented by the context of use of educational technologies (e.g., at home, in the working environment, in the classroom environment, as a “digital” source of information, as a personalised training course, as a collaborative activity), and by the proliferation of technologies becoming available for computer access (e.g., personal computer, mobile devices, public terminals, school infrastructure, real-time communication equipment, video).

Another important aspect characterising diversity is the continuous change of user-, education- and technology-oriented requirements over time. Learners and teachers
become progressively more familiar with computing technology and less reluctant to use it. Also, they refine their methods and techniques, and require less or more interaction and cooperation in different tasks and activities. New educational material in different forms and media is continuously being produced, and new mechanisms and forms of knowledge delivery and learning are likely to arise as a consequence of novel educational needs. Finally, the evolution of technology continuously brings into play new opportunities for enriching educational software in the form of component technologies, content and resource manipulation facilities, access and delivery devices, etc. As a consequence of this complex and dynamic situation, as pointed out in [Spodick, 1995, p. 2], “Tools for distance learning must be flexible and adaptable for a variety of different needs and situations, including their own obsolescence, where possible”.

2.1 HCI Design and implementation requirements

As the interactive aspects of educational software acquire progressively greater importance, HCI issues, and in particular accessibility and high quality of interaction, become critical factors in its development, and need to be carefully planned and integrated into products’ life-cycle. HCI will therefore increasingly need to provide methods and tools for constructing accessible and usable educational software. As a consequence, interaction in the emerging educational environments will have to be designed and implemented in such a way as to accommodate the diversity in the individual users, the different possible uses of the system, the technology used, the context of use, etc. This can be achieved through adaptation and personalisation.

To support the user-centred personalisation of the interactive learning process, suitable methods and techniques are necessary to facilitate the design of user interfaces capable of coping with a wide range of user- and context-related requirements in a systematic way throughout the development life-cycle of educational software, from early design phases to user evaluation. Such methods and techniques should offer adequate means to capture user- and context-oriented parameters of interface adaptation, as well as a rationale and logic of adaptation in the particular application domain, specifying how (and possibly why) a user interface should behave (adapt) in response to (changing) user- and context-oriented parameters.

Interface characteristics which can change according to these parameters include the use of traditional or novel input/output devices (e.g., cellular phone, Personal Digital Assistant), the adoption of different interaction metaphors (e.g., the classroom, the academic course) and styles, interface layouts, dialogue structures, structures and levels of system help, feedback, guidance, etc.

Another important design dimension in educational software is represented by characteristics such as high quality of use, user satisfaction, etc. Furthermore, given the highly interactive nature of educational activities, a design approach is required capable of capturing and applying different communication and cooperation models in an integrated environment, by taking into account the social context of interaction. In
this respect, educational software should be designed taking into account the need for personalised views of a shared environment, and contextual user-awareness of the processes and activities taking place within such an environment.

Finally, an important challenge in the design of educational software products is that of interaction metaphors. It is unlikely that a single interaction metaphor will be sufficient for satisfying the interaction requirements of all users with the system in a wide variety of tasks, and encourage users to take full advantage of the possibilities offered by the system. Therefore, it is particularly important that alternative interaction metaphors are investigated, new metaphors are designed and combinations of different metaphors in a unique environment are supported.

It is evident from the above discussion that coping with diversity in educational software implies the need of systematically coping with a large, complex and dynamic space of design alternatives.

From the point of view of user interface implementation, supporting adaptation in educational software raises equally challenging issues. Adaptation in the user interfaces is needed to ensure initial accessibility of the system to the end-user (adaptability). Furthermore, adaptation should support the progressive evolution of system characteristics along with the changing user requirements and preferences over time (e.g. progressive mastering of more advanced interface features, reduced need of help or guidance, etc.). Finally, during interaction, adaptation should ensure that the system appropriately responds to the changing interaction requirements of users within single interactive sessions (e.g., user tiredness, loss of concentration, etc.) (adaptivity).

As a consequence, suitable interface architectures should be employed in the development of educational software, capable of supporting the context-sensitive operation of transformations in the interface. Such transformations can be effected through the addition or removal of interaction objects, or through changes in their interactive behaviour, on the basis of the adaptation alternatives and criteria identified in the design phase, as well as on information about individual users and contexts of use. Another important architectural requirement is the capability to support the different interaction metaphors likely to be required by different users for different educational tasks and activities (e.g. classroom, book, etc).

Finally, a user interface architecture supporting adaptation in educational software should be open, i.e., it should allow for easy integration of new components so as to support the dynamic evolution and “growth” of educational software products.

2.2 Content design and implementation requirements

In Adaptive Hypermedia and Intelligent Tutoring systems, several content adaptation techniques have been developed, the most relevant of which are curriculum sequencing, adaptive navigation, and adaptive presentation. The goal of curriculum sequencing [Vassileva, 1997; Stern and Park Woolf, 1998; Stern and Park Woolf, 2000] is to provide the learner with the most suitable individually planned sequence of knowledge units and learning tasks (examples, questions, problems, etc). Sequencing
can be active or passive. Active sequencing implies a learning goal (a subset of domain concepts or topics to be mastered), and aims to provide the optimal individual path to achieve such a goal. Goals can be fixed or adjustable. High-level sequencing or knowledge sequencing determines the next learning subgoal (e.g., next concept, set of concepts, topic, lesson). Low-level sequencing or task sequencing determines the next learning task (e.g., problem, example, test) within the current subgoal. Active sequencing can be driven by the learner’s knowledge and preferences on the type and media of learning material. Passive sequencing is a reactive technique that does not depend on an active learning goal. It starts when the learner is not able to solve a problem or answer questions correctly, and aims to offer learning material suitable for filling the learner’s knowledge gap [Brusilovsky, 1999].

In the context of Web-based courseware, curriculum sequencing is a particularly important issue, given the vast amount of educational material that can be provided on the Web, and the risk that the learner gets “lost in hyperspace”.

Adaptive navigation [Calvi & De Bra, 1997; Weber and Specht, 1997] aims to guide the learner through the system by customising the link structure. Adaptive navigation determines the level of guidance and freedom granted to the student within the system, and attempts to give the learner a spatial representation of the hypermedia environment. Adaptive navigation can be considered as a generalization of curriculum sequencing in a hypermedia context [Brusilovsky, 1999]. In the context of the Web, where hypermedia is a basic organizational paradigm, adaptive navigation can be used very naturally and efficiently. Several techniques can be adopted for implementing adaptive navigation, such as adaptive guidance, link hiding, link annotation and link sorting. Direct guidance implies that the system informs the learner about the available links which point to the optimal next step to follow, established on the basis of learner’s current knowledge and learning goals. Hiding or disabling links is another frequently used technique that provides the learner with a representation of path followed through the system.

The goal of adaptive presentation [Brusilovsky et al., 1998] is to adapt the content of a hypermedia page to the user's goals, knowledge and other information stored in the user model. The “unit of presentation” is a page (or node). Pages are not static, but adaptively generated or assembled from constituent pieces. Content components may be words, phrases, paragraphs, images, video clips, etc, constituting discrete units of information about a concept. Consequently, the granularity of the adaptation may vary from word replacement to the substitution of pages or the application of different media. Content may be customised to contain additional information, pre-requisite information or comparative explanations. As a result, each learner may potentially be provided with an individually tailored course that is different to the course displayed for all other learners. Adaptive presentation is an important issue in Web-based courseware, where the "same" page has to suit the learning requirements and purposes of very different learners.

In a learner-centred perspective, content should adapt, as a minimum requirement, to the learner’s previous knowledge, individual progress, learning goals and learning style. Content adaptation should be applied each time a particular course topic is to be entered, so that the learner is initially presented with the appropriate material, as well
as during learning sessions, so that the subsequently presented material is selected on the basis of learner specific requirements at run-time. Adaptation to the learner’s navigation style and preferences should also be considered. This implies that both course structure and learning sessions are flexible and can be assembled on the fly and modified at run-time as necessary, and that the learners’ characteristics, the educational material, the course structure and the course delivery process are appropriately modelled, so that the best match between their characteristics can be established at any point in time during the use of the system.

3 Learning systems architecture

This section discusses a proposed architecture for learner-based interaction and content in Web-based educational software, comprising both authoring and delivery facilities, and in which learning resources are physically distributed. Adaptation is considered to be based on learner information, course information, learning resources information, and process information. A high-level architectural representation for such a system is depicted in Figure 1.

![Figure 1. A high-level architecture for educational software](image-url)
3.1 Learner’s model

A learner model contains explicitly modelled assumptions that represent the characteristics of the student that are relevant to the system, and in particular to the type of adaptation that it is desired to achieve [Conlan and Wade, 2000].

Several techniques are commonly used for learners’ modelling in Intelligent Tutoring Systems. One of the simplest is the use of fixed stereotypes. Learners are categorized, and adaptation is based on such a categorisation. Category assignment can be based on questionnaires or tests prior to the use of the system. The overlay model is also widely used in the educational domain. In this case, the model is constructed on a concept-by-concept basis and updated progressively. This allows, amongst other things, for a flexible model of the student's knowledge for each different topic [Brusilovsky, 1996]. For this purpose, the knowledge domain must be modularised into specific topics or concepts, and the complexity of the model depends on the granularity of the structure of the domain knowledge, as well as on the granularity of the estimation of the learner's knowledge, performed by examining the sections the learner has read and by evaluating the knowledge acquired through tests (either automatically or by traditional testing methods). Stereotypes and overlay techniques can be combined. Learners may be categorised by stereotype as an initial phase. Subsequently, the model is gradually modified as the overlay model is built from information acquired from the student's interaction with the system [Conlan and Wade, 2000].

A number of sources of information may be used to construct a learner model. The system acquires data about the user and infers learner characteristics from this data. The validity of the assumptions depends on the technique used to acquire the information. Automatic modelling may be unreliable, and therefore collaborative and cooperative modelling is frequently chosen. The learner is asked to describe pertinent characteristics directly, by filling out questionnaires and forms. Indirect feedback can be acquired from the results of exercises or problem solving tasks [Conlan and Wade, 2000].

The properties chosen to represent the user should be pertinent to the potential adaptation by the system. Learner characteristics frequently modelled are the user's previous knowledge, goals and objectives, cognitive style, learning style, maturity, general ability, confidence, motivation, preferences and background [Specht and Weber, 1996].

The domain knowledge of a student can be utilized to adapt both the content and the interface. Students' knowledge of the subject evolves continuously, and systems need to adapt to the changing knowledge state of a user. Depending on the knowledge of the learner the presented content can be restricted or expanded. Additional explanations can be given to students with more sophisticated knowledge while novices are confronted only with the basic information. The knowledge state of a student can be relevant for a number of adaptation types, such as, for example, the selection of the next task (task sequencing), the number of tasks (mastery learning) and the next learning intervention, and the adaptation of questions and explanations.
Learners’ goals can be distinguished into high-level and low-level goals. A high-level goal is stable during a session or across sessions (e.g., the acquisition of knowledge in the subject domain). Low-level goals are subject to changing often and quickly (also within a single session). Low-level goals, like learning something about a certain domain concept or solving a problem are often important for short-term adaptation purposes (e.g., displaying or bringing to focus relevant parts of content). Information about a user’s goals and interests is also an important source of information for choosing the form of presentation of selected content [Specht and Weber, 1996].

The provision of learner-centered adaptation in Web-based courseware should capture all information considered relevant regarding the engagement of learners in electronic training courses. In order to ensure wide applicability and reusability, the model should be general enough to address different categories of knowledge domains, learning purposes and training processes. For example, it should be possible to create learner profiles for formal classroom education, professional training, informal learning, etc. This implies the need of differentiating the type of information to be modeled according to its relevance for the specific case. For instance, in classroom-based training, evaluation records may need to be maintained, while in professional training, organizational-role specific information will need to be managed. To achieve such a flexibility, the learner model should be based on meta-data, and encompass appropriate abstract attributes, enabling concrete learner models to be instantiated for different knowledge domains and training processes. It is important that meta-data take into account currently proposed standards [IEEE-LTSC, 2000b; IMS, 2001a]. As intensive use of modeling is required, models should be easy to create, consult and up-date. A possible technique for model information input is the provision of alternative templates for specific domains. Query facilities for existing profiles should also be provided.

For interaction adaptation, knowledge about relevant learners’ characteristics is also necessary. These may include sensory, motor and cognitive abilities, expertise in the use of computers, of the Web, and of the particular system, knowledge about specific interface components, colour and media preferences, preferences in interaction and navigation styles, etc. Different educational methods and styles also need to be supported (e.g., group learning, instructional or constructive learning, peer tutoring).

3.2 Course model

Besides information about learners, an appropriate model of the data to be delivered in a course is necessary, as well as of the delivery process. Content information, constituting the “raw material” which is to be appropriately selected and dynamically assembled in a learner-adapted course, should be logically separated from the course information, providing semantic information regarding the overall course structure. This logical split enables a course to be delivered in different forms, by controlling various semantic parameters, such as level of detail, type of explanation, and complexity of examples. These models should describe all aspects of how the content
should be used and whom it is designed for. In this section, the first model will be referred to as content model, while the second will be referred to as process model.

The content model identifies: (i) the elementary constituent units of content; and (ii) the structure in which elementary constituents are organised. In recent standardisation efforts, units of content are referred to as ‘learning objects’ [IEEE-LTSC, 2000a]. Such a unit can be a paragraph or other piece of text, an image, a video clip, etc. The unit internal structure should not be relevant for the model. Units may be static (stored) or may be generated by an external application (like a natural language generation module) [De Bra, 1999].

The knowledge domain can be described in the content model in terms of concepts (or terms) and relationships between concepts. Concepts may be composite, and can be used for defining prerequisites providing meaningful paths through the information. When concept A is a prerequisite for B it means that the user should visit (pages about) A before B. Other types of concept relationships can be defined as well. An example is inhibitor relationships: A inhibits B means that after a learner has visited concept A, it is no longer necessary that concept B is visited [De Bra, 1999].

The granularity at which the content is stored determines the level of content adaptation that may be achieved, and richer metadata sets are required to adequately describe finer-grained content. Extra level of detail in the metadata can be required, for example, to distinguish similar, but different units content. For example, two pieces of content can both describe the same concept. One may be a theoretical description, while the second may be an application example. The model should offer a clear way to differentiate these two objects [Conlan and Wade, 2000]. Reuse potential is also closely related to the granularity of the content description. Coarse-grained content may be too domain specific to be easily reusable.

In learner-centred adaptation of Web-based courseware, the notion of learning object is particularly important. Learning objects should link directly to content, and encompass course-related semantic attributes. As content can manifest itself in different forms, learning objects should not have a singular mapping to content resource items (i.e., content entities). Rather they should be abstract, i.e., capable of linking to different content items. Additionally, they should not be physically attached to content. This can be accomplished by associating learning objects to content via resource identifiers, as opposed to embedding content within learning objects. Such a logical split also support a greater degree of reusability. Learning objects do not need to be restricted to simple elements such as paragraph items, but may range from multimedia content, more comprehensive instructional content, to external software and relevant tools. In a wider sense, learning objects in Web-based courseware could include, and are not limited to, passive items, active items, collaboration gateways, personal contacts, and social events. Alternative templates can be supplied for these meta-data categories. Some typical attributes of learning objects could include, for example: type, Author, Owner, Terms of distribution, Format, Teaching / instruction style, Grade level, Mastery level, Prerequisites, etc. For interaction adaptation, learning objects should also include information concerning their interaction characteristics (e.g. color, font, interface objects, required interaction techniques), to be matched with learners’ interaction requirements and preferences.
Current proposed standard specification can be considered as a comprehensive modelling of learning objects, aiming to be domain-free [IEEE-LTSC, 2000a]. In this respect, it constitutes a universal template. However, due to its intended generality, such a specification offers a taxonomy of semantic attributes which do not allow for domain-specific concepts, something which could constitute serious constrain domain-oriented adaptations. Therefore, domain specific templates should also be defined.

The structure of a course may vary depending on many factors. For instance, educational material for a course may be structured as a hypermedia book, a tutorial, a training session, a seminar, etc. Course structures are usually specified on the basis of a hierarchical organisation model, in which the most primitive constituent components are the learning objects. Standardisation proposals are being elaborated in this respect also. The Content Packaging Information Model [IMS, 2001b] describes a self-standing package of learning resources.

From the point of view of adaption, the process model constitutes the main backbone for matching the content to be provided to the learners’ characteristics. Different templates can be provided for different type of courses. An important requirement on course templates is that they cater for alternatives to be made available to different users in different situations, and provide information for selecting amongst alternatives, i.e., they define multiple courses in one hierarchical structure, depending on the learner’s level, purpose of learning, background, acquired prerequisites, etc. Learning objects’ templates and course templates should appropriately link, in order to enable the implementation of facilities such as, for example, search, evaluation, selection, and acquisition of learning objects, sharing across different courses, semantic manipulation in creating personalised learner-centred lessons, and documentation and recognition of completion for learning / performance objectives associated with learning objects. The process model should also provide information for course management at run-time (e.g., start date, expected duration, end date, real duration, delays causes, course identification, sessions, plan, enrolled learners, tutors, course topic index, completion status, feedback from learners).

### 3.3 Adaptation Logic

In Adaptive Hypermedia and Intelligent Tutoring Systems, adaptation decision-making is often embodied in the “pedagogical knowledge” of the system and is implemented through rule-based approaches [e.g., Vassileva, 1997; De Bra et al., 1999]. Alternative approaches include Bayesian networks for calculating probabilities [Henze, 2000], fuzzy logic [Nkambou, 1997], and neural network and connectionist methods [Mullier et al., 1999].

In the proposed architecture, the adaptation logic provides the reasoning mechanism required for accomplishing both interaction and course adaptation, on the basis of the relevant characteristics captured in the models. The reasoning process, based on the information contained in the learner and course model, will take place both at the beginning of and during each learning session, and drives to decisions
concerning the learning objects which should be displayed and how they should be assembled. In other words, the adaptation logic provides a decision mechanism for dynamically selecting, among the multiple instances of course components and learning objects available, the one(s) appropriate for presentation at a given point in a learning session of a specific user. In this respect, the adaptation logic combines the information provided in all models in a coherent whole. Potentially suitable implementation techniques for adaptation logic include production systems, decision trees and declarative logic programming. Although general adaptation rules can be defined, it is unlikely that all adaptation cases for a variety of learners’ characteristics and courses can be captured at a general level. Therefore, adaptation rules should be editable, i.e., course authors should be allowed (and supported in) defining their own adaptation logic on the basis of the knowledge domains, learners and purposes their course is designed for.

Apart from deciding which learning objects should be supplied to learners, the physical form of such objects should be retrieved and delivered appropriately. Delivery may imply the presence of mechanisms to assemble a physical structure that is not only “acceptable”, but also high-quality. For instance, assume that the learning objects correspond to independent paragraphs in a document, while the necessary physical data are retrieved from different locations, and may contain both text and images. The assembly of those items into a single document should be carried out in a way leading to a high-quality visual document structure. Additionally, the delivery platform should be able to handle the retrieval and display of such physical data (i.e., text and images in their particular format). The content assembly process can be based, for example, on domain-specific HTML templates.

3.4 Delivery mechanism

In Web-based systems a clear separation between the content and the user interface is not possible. The reason is that content is also interactive, supplying all kinds of interaction objects and techniques which are normally classified as belonging to the interfacing layer. Therefore, two types of user interface can be identified: the delivery interface, and the content interface. The former provides standard facilities such as history control, navigation, and communication, while the latter provides content-specific interaction methods, which depend on the physical delivery of a learning object (i.e. embedded in form). Consequently, to accomplish user interface adaptation, both levels of interfacing need to be affected. The delivery interface can realise adaptation according to information on learners’ interaction requirements and preferences. Regarding the content interface, meta-data specification for learning objects need to attach interaction-related attributes to the physical resource identification, thus supporting the selection of the most appropriate physical forms.

Adaptations may concern the lexical, the syntactic or the semantic level of interaction. More specifically, adaptation may be used to facilitate the presentation of interactive objects. Attributes that are subject to adaptation include size, colour combinations, font family and size, placement, grouping, etc. Adaptations at this level
may also relate to physical elements of a user interface, such as the topology (e.g., horizontal, vertical), the presentation / output device, etc. The use of multimodal, usually redundant, representations, and of suitable interaction techniques, engaging combinations of user modalities (e.g., visual and auditory) constitutes another area where adaptations can be applied.

Furthermore, adaptations may address the interaction syntax in a user-system dialogue. This entails determining, and engaging the user in, a suitable sequence of interaction steps required for the successful and efficient completion of a task or activity. Indicative adaptations at this level include adaptive requests for confirmation, selection of suitable command order (e.g., object-function versus function-object syntax), alternative function activation modes (e.g., explicit versus implicit activation), multiple levels of help or guidance, etc. In addition, several techniques may be used to augment navigation and facilitate performance targets (e.g., effectiveness, efficiency and satisfaction). For example, various tools for guided navigation (e.g., structured views, maps, etc.) may help reduce the need for backtracking in the course structure and thereby improve efficiency. Moreover, additional functionalities (e.g., use of redundant or special interaction techniques) may be used to anchor users in stable contexts. This would minimise disorientation and distraction and would allow users to focus their cognitive resources on the task at hand, rather than the overall system’s use and navigation.

Finally, adaptations may be targeted at the interaction semantics. This relates to what the user perceives as the prevailing interactive embodiment (interaction metaphor) of the computer-based environment (e.g., desktop, book, rooms) that materialise the educational / knowledge delivery space, as well as the functionality that should be available. Adaptations at this level mainly concern the metaphor(s) used to embody different functional properties of the system. For instance, using a metaphor to develop a suitable visualisation of a collection of material for a specific course is an example of embedding metaphor in the user interface. In contrast, developing an interface which allows the user to interact with educational software through school-related rather than desktop-related concepts such as a class-room, library, laboratory, cafeteria, etc., implies using metaphors to characterise the overall interactive embodiment of the computer-based environment. Adaptation may be used not only to select and instantiate suitable interaction metaphors for different users, technology, contexts or even specific tasks, but also to individualise the interaction and communication process.

It is also important to note at this point that the capability to adapt a specific aspect of interaction, does not necessarily imply the necessity, or, for that matter, the usefulness of doing so. Rather, each such adaptation capability should be treated as a tool in accommodating specific design problems (diversity and universal accessibility being amongst the most prominent ones), and should be treated with care, as exaggerations in the employment of adaptations could significantly compromise the usability, user-friendliness and overall acceptability of the system.
5 Summary and Conclusions

Current and emerging trends in the field of educational software move towards the direction of supporting learning for anybody, anywhere, at anytime in the emerging information society. In this respect, accessibility and quality in use of educational software are becoming compelling and challenging issues.

This paper has discussed some important aspects of a proposed architectural framework suitable for supporting interaction and content adaptation in Web-based courseware. The adopted perspective is that adaptation should be learner-centred, i.e., it should continuously deliver, at any time during the use of the system, the most appropriate interactive learning content to each individual learner. Adaptation has been considered at both the content and the interaction level. A first implication of this is that content adaptation should take place not only on the basis of the educational and information characteristics of content units, but also on the basis of the interaction properties of such units. Secondly, this implies that content should be dynamically assembled in a way that produces high quality results in terms of both appropriateness to the learner’s knowledge status and learning goals, and appropriateness to the learner’s interaction requirements and preferences. As a consequence, the adaptation of the interaction should be investigated with respect to its relationships to the adaptation of educational content and content delivery method, so as to lead to a global systematic approach toward learner-centred adaptation. This implies identifying the required modelling attribute in the learners’, educational content and delivery process models, as well as their interrelationships. While these models and meta-data should build on existing standard proposals as well as on previous work in the fields of Intelligent Tutoring Systems, Adaptive Hypermedia and Human-Computer Interaction, they also need to be general and flexible enough to cater for a variety of knowledge domains and educational purposes, and at the same time to be specific enough to support meaningful adaptation in each application. Furthermore, models should be simple enough to allow extensive modelling on the part of course authors and teachers, and should support the easy definition of alternatives for course structures, educational material units and interaction behaviour characteristics of such units.

In this context, several research issues become relevant. Firstly, an effective use of adaptation in user interfaces of software products implies a deep and thorough knowledge of the users as individuals and as a community, of their requirements, and of how these requirements map to concrete designs for adaptation at the lexical, syntactic and semantic levels of interaction. Design principles and guidelines have to be elaborated to cater for the universal accessibility and high quality of interaction of educational software. This includes the investigation of the social and cooperative aspects of educational software under an adaptation-oriented perspective, the identification of what types of adaptation are necessary to support cooperative learning, and how they should be applied. In an analogous way, research efforts are necessary towards the investigation of the environmental and institutional context of adaptation. It is likely that completely automated assessment of the user progresses is not a viable solution, and that human assessment is required at some point in the
process. However, on-line monitoring of dynamic user characteristics, which may change in the time-frame of a session, is also required. Therefore, learner characteristics should be carefully investigated with respect to when and how their values and history need to be updated.

References


