

---

# CHAPTER 2

## PERSONALISING WEB-BASED LEARNING

---

### 2.1 INTRODUCTION

This Chapter presents a characterisation of some of the main issues driving personalisation of learning. A reflective account is provided that considers the recent evolution of both learning theories and educational applications of technology that have come to shape modern learning technology and its approach towards personalisation.

A critical analysis of the published literature related to personalisation techniques and adaptivity for learning is also carried out. The general structure and functionality of Adaptive Hypermedia (AH) systems are introduced and their underpinning models explained. A series of AH systems for learning are reviewed in terms of these underlying models. This Chapter also puts forward the use of cognitive styles for the design of adaptive Web-based learning systems.

### 2.2 DRIVING ISSUES BEHIND PERSONALISATION OF LEARNING

Education stands on the edge of a revolution through the use of ICT [Jenkins & Hanson, 2003]. Increasing and varied pressures driving the greater use of e-learning within the sector can be found at the global level in terms of economic, social and cultural demands towards a knowledge-based society [UNESCO, 2004]. Pressures exist relating to quality assurance, enhancement of teaching and learning, increasing access, widening participation and improvement of efficiency [Fry et al., 2003, Jenkins & Hanson, 2003]. Increase in student numbers, particularly in HE, places further pressure on resources and demands greater flexibility on modes of study and delivery [Fry et al., 2003].

Student needs and expectations are also changing. Today's students have grown up surrounded by technological devices and increasingly come to the education institutions expecting to use ICT for learning [Prensky, 2001]. Expectations about technology are also high among adult professionals who need access to formal education throughout their careers and count on e-learning for their continuous professional development

[Miller & Schiffman, 2006]. Employers also expect their staff to cope with the modern technology-based work environment.

In addition, current approaches to teaching and learning are moving towards a learner-centred model that recognises that each student approaches learning from their own perspective [Laurillard, 2002; Fry et al., 2003]. Technology is expected to help provide the appropriate learning opportunities for students from a wide range of backgrounds and levels of abilities.

In this general context, ICT is seen as a viable alternative for improving cost-effectiveness of educational resources and increasing access to learning and training, as well as enhancing attention to diversity and quality. It is plain to see that supporting the learning needs of students at the individual level is a key issue within this framework.

This research focuses on the provision of adaptive learning materials to support student learning, and considering the wide range of technologies available for this purpose, its scope is limited to Web-based learning environments.

### **2.3 FROM LEARNING TO WEB-BASED LEARNING**

Different theoretical and technological approaches have contributed to the development of learning technology. Learning theories, for example, are gradually moving from a teaching-centred to a learner-centred model. Educational uses of technology are also changing, increasingly enabling pace, sequence, and instructional strategy to fit each learner's needs [Wiley, 2003].

For the purposes of the research, it is important to understand how learning technology has changed during the past years and how its theoretical underpinnings have evolved as our understanding of the learning processes evolves. We might then try to identify what the requirements are to move personalisation of learning forwards.

Considering that different terminology is used to describe similar concepts, a useful starting point seems to be the identification of some terms that may be open to different interpretations and to outline the context in which they are used in the research.

Learning Technology (**LT**) is referred to as the application of technology to facilitate and enhance teaching, learning and assessment [Rist & Hower, 1995; Seale & Rius-Riu, 2001]. LT includes, but is not limited to, computer-based learning, multimedia materials, and communication systems in either face-to-face or distance contexts.

In the past, the wide range of computer applications regarded as LT has been classified under various acronyms (Table 2.1), such as CAI, CAL, or CBL [Rist & Hewer, 1995]. As technology evolves new terms are continuously introduced, for example Computer Aided Assessment (**CAA**), or Computer Mediated Communication (**CMC**). The same is true for technology aiming at supporting the management of the learning experience, such as Virtual Learning Environments (**VLE**), Managed Learning Environments (**MLE**), or Learning Content Management Systems (**LCMS**).

CAI	Computer Aided Instruction
CAL	Computer Aided Learning
CBL	Computer Based Learning
CAA	Computer Aided Assessment
CMC	Computer Mediated Communications
VLE	Virtual Learning Environments
MLE	Managed Learning Environments
LCMS	Learning Content Management Systems

**Table 2.1:** Acronyms used to refer to learning technology. A wide range of applications have been classified under various acronyms.

Over the past few years, the advancement of the Internet and the Web has encouraged its adoption as a learning infrastructure to organise, create, enable, deliver, and/or facilitate learning. Accordingly, the terms online learning and Web-based learning refer to electronic means of distributing and engaging with learning, typically via the Internet and related media services [Jolliffe et al., 2001; Stephenson, 2001]. This infrastructure facilitates access to and delivery of coordinated collections of learning materials, also enabling different kinds of interactions between learners and teachers.

More recently the term e-learning has emerged as a synonym of technology-based learning. It covers a wide set of applications and processes, such as computer-based learning, Web-based learning and digital collaboration, as well as a variety of electronic media such as the Internet, intranets, extranets, satellite broadcast, audio/video tape, interactive TV, and CD-ROM to support teaching and learning.

A difference is noted between e-learning and Web-based learning, since the latter constitutes a part of e-learning and describes learning via Internet, intranet, and extranet.

With the understanding that Web-based learning is a subset of the overall LT scope, the research focuses on the facilities provided by this medium to personalise the presentation of learning materials.

### **2.3.1 The Influence of Theory**

Laurillard [2002] suggests that “the promises made for e-learning will only be realised if we begin with an understanding of how students learn, and design the use of learning technology from this standpoint” [Laurillard, 2002, p. i]. In recent times, three theoretical approaches have been especially influential in the design and development of LT, namely Behaviourism, Instructional Design, and Constructivism.

#### **The Behaviourist Approach**

Behaviourism considers that human learning is observable in their behaviour, and that behaviour is shaped by the environment. Accordingly, learning can be conditioned by the interaction of stimuli (manipulating the environment) and response, where association and reinforcement are important. Learning is therefore considered as the acquisition of new behaviours [Hills, 1982].

This approach has strongly influenced education in the form of learning outcomes, i.e. specifications of desired outcomes in terms of measurable behavioural targets shaped by a series of reinforcements on intermediate steps [Boyle, 1997]. Teaching machines and the model of programmed instruction were early realisations of Behaviourism.

#### **Instructional Design**

In the mid sixties, Robert Gagné published his book “The conditions of learning” [Gagné, 1985], where he laid the basis for Instructional Design, an approach that is now regarded as a principled method for generating teaching strategies.

Gagné identified general types of human capabilities that are learned, such as intellectual skills, cognitive strategies and verbal information. Next, each capability was analysed to decompose it into a sequence of “learning events”. Finally, the combinations of capabilities and learning events were synthesised to guide the design of the instructional events to be carried out by the teacher [Gagné, 1985].

In practice, Instructional Design represents a prescriptive method carried out through three main steps: (1) needs analysis, (2) selection of instructional methods and materials, and (3) evaluation [Boyle, 1997; Laurillard, 2002]. This guidance is largely media independent and can be used for the development of a wide range of learning events. It has been particularly influential in the design of computer-based learning.

## **Constructivism**

Constructivism states that individuals build their own intellectual structures through interacting with their world [Papert, 1980; Laurillard, 2002]. Learning is an active process of constructing rather than acquiring knowledge, while teaching is a process of supporting that construction rather than communicating knowledge [Laurillard, 2002]. Therefore learning, rather than teaching, is primary.

Two different trends can be identified: “cognitive constructivism” and “socio-cultural constructivism”. The former comes from the work of Jean Piaget (Swiss, 1896-1980) describing how children (and indeed adults) are continually generating knowledge and developing increasingly abstract constructions about the external world [Papert, 1980; Mayes, 1998; Fry et al., 2003]. The latter derives from the work of Lev Vygotsky (Russian, 1896-1934) describing the development of knowledge through social interaction and the idea of “situated cognition” [Laurillard, 2002].

Currently, Constructivism has become the most influential approach since its strong theoretical base offers greater scope for exploiting the opportunities offered by modern LT. In contrast to the formal and precise approach of Instructional Design, Constructivism offers some basic principles that inform the design of learning environments. Boyle [1997] offers a useful summary of these principles in terms of:

- Providing authentic learning tasks
- Developing interactive social contexts for learning
- Allowing students greater responsibility over their learning experience
- Encouraging learners to become proficient in the process of constructing knowledge
- Promoting self-awareness of the knowledge construction process

### **2.3.2 The Case for Technology**

LT has also evolved during the past years alongside learning theories, although not always based on the same principles. Indeed some researchers [Jolliffe, et al., 2001; Stephenson, 2001; Laurillard 2002] have observed that more LT, and particularly computer-based applications, appear to have been developed because it was technically possible to do so, without explicit reference to any pedagogical principles. It might be argued however, that while technology itself does not equate to an effective learning

environment, its implementation has generally sought to enhance teaching and learning, and in so doing there has usually been an implicit or explicit model of learning behind.

Since LT has been used to support a variety of learning settings, a comprehensive review of these would be outside the objectives of the research. Some approaches that have been particularly influential in shaping the modern LT have been identified instead; some are complementary rather than conflicting approaches, and some others overlap, thus the division used below is primarily for descriptive purposes.

### **The computer as a tool**

The notion of intelligent machines for teaching and learning can be traced back to the 1920s. These were originally mechanical devices that presented instructional content to students in a pre-defined order, allowing advancement only as the correct answer was given. Later on, during the late sixties, the rigid paths of early machines were replaced with computer-based branching systems and the provision of some feedback about learning performance [Boyle, 1997].

As computer technology developed, so did the range of its uses. For example, software applications such as word processors, spreadsheets and statistical and qualitative data analysis tools have been used to develop students' skills, ideas, hypothesis and understanding, and to present results and findings. Software applications also exist that support the learner in the formation of new concepts, such as concept mapping tools.

Other kinds of software tools for learning include applications that support educational processes defined independently of the tool itself, for example, graphical editors used to help engineering students to present their design ideas. Visualisation tools used to represent complex sets of data in a visual way also fall within this category; examples include weather systems, topography, financial systems and medical imaging.

### **Adaptive Media**

Computer technology can be used to support vicarious experience. Applications that use different modelling strategies to accept and transform input from the user are particularly well suited in this category. Simulations, for example, are designed to model reality. Often a real-life situation may be dangerous or too complex to be used in an educational setting. A simulation can focus on specific aspects of the situation and provide a safe environment for the learning experience [Hills, 1982]. Depending on the

level of interactivity allowed, their educational utility may be confined to motivation or engagement, as in the case of simulations where the participation of the user is limited to observing the operation of a particular process [Boyle, 1997]. When the level of interaction allows for the provision of direct intrinsic feedback on the user's actions, simulations may help students to relate theory and practice [Laurillard, 2002].

Computer educational games are usually simulations of reality which introduce the competitive element [Hills, 1982]. Participants can interact within a simulated environment, and thus learn through experimentation. Games can also be used to promote engagement.

While simulations model the behaviour of a system, virtual environments use a graphical model to display the visual properties of a system. Examples would be virtual art galleries or virtual field trips. Recent developments in this area show that augmented reality can be very effective in training complex skills, for example surgery.

### **The Logo Approach**

During the late 60s/early 70s, beyond the strong influence of Instructional Design, Seymour Papert advocated a different approach to computer support for learning. He proposed a series of principles for the design of learning environments that would facilitate the process of cognitive development. These principles are embodied in Logo and the interactions it supports [Papert, 1980].

Logo is a programming language for geometry. Its best-known feature is Turtle Graphics, a turtle represented by a cursor on the computer screen or, alternatively, by a small floor robot. A program is a set of instructions the child gives to the turtle to do something, such as drawing a rectangle or a triangle. In a subsequent program the child may combine these basic shapes (procedures) to create more complex figures, such as a house. As a transitional object, the turtle allows the children to make sense of mathematical tasks in terms of everyday familiar experiences [Boyle, 1997]. In this way, an environment is created where children actively explore abstract concepts; these environments are called *Microworlds* [Papert, 1980].

Contrary to most approaches where learners are taught by computers, in Logo the child programs the computer, and in teaching the computer they explore how they themselves think. Thus the process of learning is transformed: it becomes more active and self-directed, knowledge is acquired for a recognisable personal purpose [Papert, 1980].

Currently, multimedia technology allows the construction of much more sophisticated microworlds than can be created in Logo. However, its theoretical foundations are still valuable for developing interactive applications and environments for learning.

### **Hypertext and Hypermedia**

In its original form, the aim of hypertext was to go beyond the static and sequential form of plain text to a more dynamic structure with the possibility of creating links between parts of the document or between different documents, allowing also the inclusion of references for later retrieval [Sloane, 1996; Shneiderman, 1998].

During the eighties a number of commercial hypertext systems became available; among those, HYPERCARD software for Macintosh computers had a great impact. It combined hypertext features with the graphical user interface of the Apple computer, and importantly: it allowed the user to create their own associations and build their own information environments including buttons and other fields, with no knowledge of programming necessary [Boyle, 1997; Shneiderman, 1998; Laurillard, 2002]. As Logo, HYPERCARD was considered a programming environment which illustrated how people could learn by teaching the computer, rather than the other way round.

With the advancement of technology hypertext transformed into hypermedia, giving access to navigable links between objects of different media types [Sloane, 1996; Shneiderman, 1998]. Stand-alone educational hypermedia resources take the form of CD-ROM or DVD-based encyclopaedias and electronic reference books. Their main feature is the large range of material they make available for exploration: text, pictures, graphics, audio and video together in one disc. As a result previously time-consuming research tasks become more feasible.

### **2.3.3 Online Learning**

The advent of the Internet for educational purposes brought about a wide range of applications developed mainly to automate existing practices. Games and simulations were made available online, information retrieval was enriched with the availability of large collections of materials over the Web, lectures were made available online, and online learning materials started including audio and video in addition to text.

Some other learning events were made possible, in particular the use of the Internet to facilitate communication between students and between students and their teachers.

Electronic synchronous and asynchronous communication appeared, as did strategies ranging from online discussions and bulletin boards, to online debates and role-play simulations, incorporated as part of the learning process [Alexander & Boud, 2001].

Information systems have also been developed to facilitate learning and the management of that learning. VLE, MLE and LCMS are some examples of these kinds of systems. These are pieces of software, commonly accessed via Web browsers, which supply an integrated online learning environment to support learning and teaching. Commonly, these include tools for content delivery, tracking of student activity, communication and assessment [Jenkins & Hanson, 2003].

In general, different modalities of online learning are currently identified, such as Web-facilitated courses, where hand-outs, syllabus and assignments, for example, are posted using web pages or content management systems. There is also a blended approach, where face-to-face interactions are combined with content material delivered online, as well as collaborative activities using tools such as discussion boards. When applied to distance learning, Web-based technology is used to deliver most of the course content, with few or no face-to-face meetings, but with collaborative activities and electronic communication between tutors and learners [Allen & Seaman, 2004].

Irrespective of the online learning modality adopted, some aspects that are commonly highlighted include content delivery in multiple formats, collaborative interactions for learning and learning support. In turn, these elements seem to be the result of an increasing agreement on the theoretical underpinnings of learning environments (online and otherwise) [Mayes, 2001], i.e. a shift to a constructivist approach in which learning is developed by the individual through activity, and a change towards supporting this process both in the social context and individually. Activities and materials must be designed that encourage students to build on what they already know, to actively reflect on their own ideas and experience, and to interact with peers so they can make use and extend the information acquired [Alexander & Boud, 2001].

Several issues can be highlighted from the review carried out through the previous sections. First, approaches to the use of LT have evolved under different educational principles and technological advancements and constraints. Second, never before has there been so much agreement about the pedagogical principles guiding the use of LT and particularly Web-based learning [Mayes, 2001]. Third, approaches to realising the principles of Constructivism are diverse, most of them with advantages and disadvantages depending on the characteristics of specific learning settings. Delivery of

learning materials is, however, central to most of them. Finally, while the social interaction for learning is currently an important concern, there is still scope for improving learning support at the individual level.

Taking these elements into account there is a clear need for supporting different approaches to Web-based learning while providing support to the individual learner. A suitable approach in this direction seems to be the personalisation of Web-based learning materials based on the individual characteristics of learners.

## **2.4 DEALING WITH INDIVIDUAL DIFFERENCES**

Learners are not all the same. While physical differences such as height, facial characteristics or tone of voice are obvious, other differences are also apparent, such as the way people behave, think or learn [Dix et al., 1993; Jonassen & Grabowski, 1993; Riding & Rayner, 1998]. One of the main concerns in the research is to identify some of those differences that may be useful for personalising Web-based learning materials.

To say that learners with different characteristics will not respond similarly to each form of instruction may seem obvious. It may also seem obvious to presume that if LT is used by a wide variety of learners, a design approach should be applied that takes these differences into account. Until recently however, the notion of the “average user” has promoted a “one-size-fits-all” approach for the design and development of computer applications, LT included. This approach is gradually changing and, given the fact that computers can be used to monitor the interaction at a level of detail unavailable to other artefacts, computer systems are increasingly being supplied with the capability of changing their functionality, structure and/or representations to better match the preferences, needs and goals of users [Benyon, 1993; Brusilovsky et al., 1996, 2001].

Adaptable (or customisable) systems have built-in flexibility to allow the user to select among different options that will alter the presentation, content or functionality of the application [Benyon, 1996; Cristea & De Bra, 2002]. A simple example of this approach is to have the option of changing the colours displayed on the screen of our computer. Adaptivity, on the other hand, is referred to as the actual capability of a system to automatically adapt to new conditions – usually derived from an underlying model [Benyon, 1996].

## 2.4.1 Approaches to Personalisation

Embedding adaptable and adaptive features into LT aims at increasing its functionality to cater for the needs of learners by making applications personalised. The term personalisation refers to the process of tailoring a product or service specifically to one individual (or a group of individuals) [Bental et al., 2001].

Currently, many services on the Internet offer some level of personalisation to allow users to receive information and services according to specified preferences. Tailoring may be based on implicit data, such as items purchased and pages viewed or explicit information provided by the user in the form of ratings or preferences. Information is transformed by means of content selection and filtering, changes to the structure of the content, or the adoption of different presentation styles [Bental et al., 2001].

Some dimensions have been suggested to analyse personalisation, such as input factors, data collection methods, reliance on predictions, user involvement and timing [Bental et al., 2001; Cranor, 2003]; these are explained in more detail below.

### **Input factors**

- *Individual information* – Personalisation is based on the individual's demographics, preferences, knowledge or behaviour, which may be observed or otherwise recorded.
- *Collaborative factors* – In collaborative models, information about several other users is gathered to compare it with and/or make inferences about the current user.
- *External information* – External or environmental factors which are not directly part of the user model but are relevant to specific users, for example, season of the year, current time, type of browser, system characteristics or connection speed.

### **Data collection**

- *Explicit data collection* – Information used as source of personalisation is explicitly provided by the user. For example, some systems allow their users to create personal pages or customize their view of a site based on their personal preferences, some other applications require the user to rate a number of items provided.
- *Implicit data collection* – Personalisation is based on information inferred or predicted about the user. Interests or preferences are built by observing the user's

behaviour – for example, search queries, purchase history or browsing – and using some learning mechanism.

- *External data collection* – External sources of information are used for personalisation – for example, a patient’s medical record.

### **Reliance on Predictions**

- *Content-based personalisation* – User’s explicit stated preferences are used to transform the output presented. Specific requests or other actions can also be used to trigger automatic personalisation; for example, if a user buys a book, a website may suggest other books on the same topic.
- *Prediction-based personalisation* – Profiles are built based on the user’s explicit or inferred ratings, which can be compared with the profiles of other users. These models may base their predictions on fixed stereotypes of different users’ behaviour, or they may use learning algorithms to adapt constantly to their users.

### **Timing**

- *Single or occasional use* – Some systems provide personalisation based on information provided by or inferred from the user for one session or while completing the current task.
- *Repeated or Frequent use* – Many personalisation systems are intended for repeated use. These commonly develop profiles of users and add explicitly provided or inferred information about them each time they return to the site. Services such as daily news or stock updates are examples of this category.

### **User involvement**

- *User-initiated personalisation* – Applications of this kind offer users the option of selecting some kind of personalisation. Users might, for example, select their preferred layout or the number of items to be displayed; they might also ask for regional information or items of interest.
- *System initiated personalisation* – Personalisation may be offered without the user realising, through tracking activity or knowing who the users are through some form of authentication.

## 2.5 ADAPTIVE HYPERMEDIA SYSTEMS

While personalisation is one of the most important areas of research in computing, its development is largely based on the foundations provided by Adaptive Hypermedia (AH). AH integrates knowledge from hypermedia systems development and user modelling aiming to provide personalised functionality to the individual user. AH research can be traced back to the early nineties, when a number of research teams started exploring ways to adapt the output and behaviour of systems to individual users [Brusilovsky, 1996, 2001]. The arrival of the Internet and the Web represented a turning point since it prompted interest in offering personalised services based on this technology [De la Flor, 2005]. From then on research and development relating to adaptation and personalisation of Web-based applications has increased considerably.

Brusilovsky [1996, 2001] identifies six main types of AH systems:

- Educational hypermedia
- Online information systems
- Online help systems
- Information retrieval hypermedia
- Institutional hypermedia
- Personalised views in information spaces

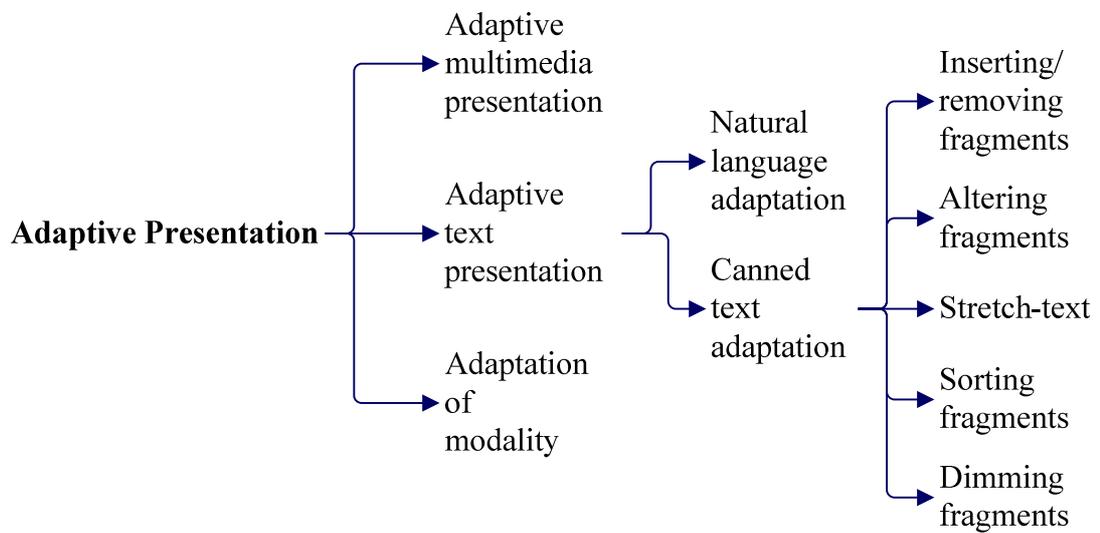
### 2.5.1 Adaptation Methods and Techniques

There are two main areas of adaptation in AH systems: adaptation at the content level or *Adaptive Presentation* (Figure 2.1), and link level adaptation or *Adaptive Navigation* (Figure 2.2.) [Brusilovsky, 1996, 2001; De Bra et al., 2005].

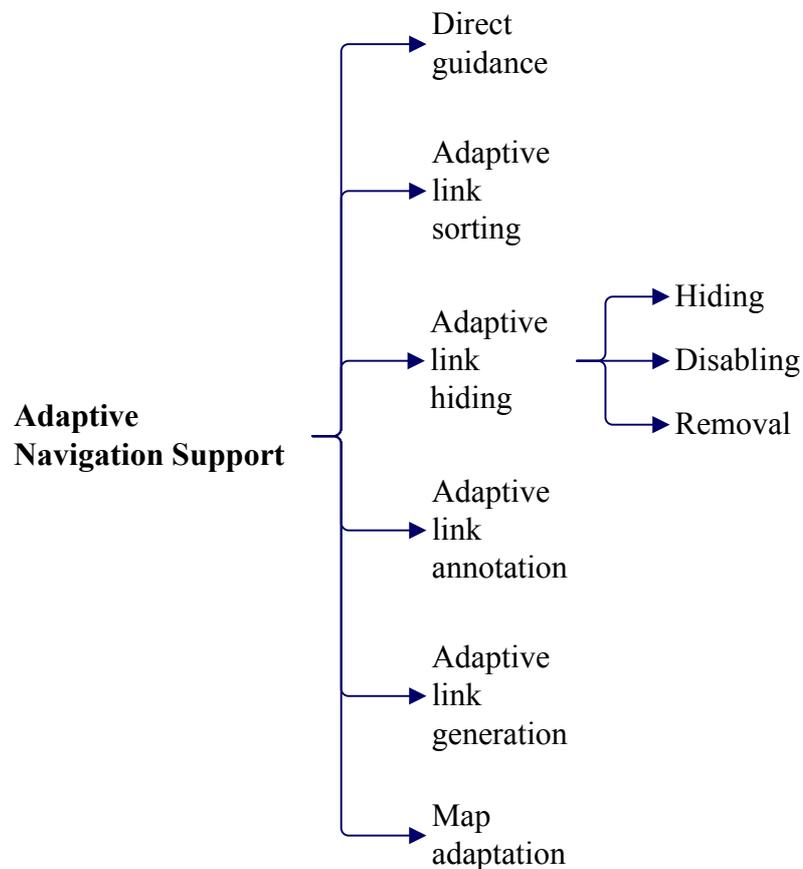
#### **Adaptive Presentation**

Adaptive presentation comprises a series of techniques used to assemble the content of a webpage from smaller components according to the current state of a model of the individual user. In this way, different users accessing the same webpage may be presented with different content. Techniques proposed include various approaches for inserting, removing or sorting content fragments associated to different conditions that determine their inclusion in the web pages presented. Conditioned fragments can be

implemented, for example, in the form of pre-requisite explanations, additional content or comparative explanations. Other techniques have been developed to assemble content including media elements of different types depending on its relevance to the user.



**Figure 2.1:** Taxonomy of adaptive presentation technologies [Source: Brusilovsky, 2001, p. 100].



**Figure 2.2:** Taxonomy of technologies for adaptive navigation support [Source: Brusilovsky, 2001, p.100].

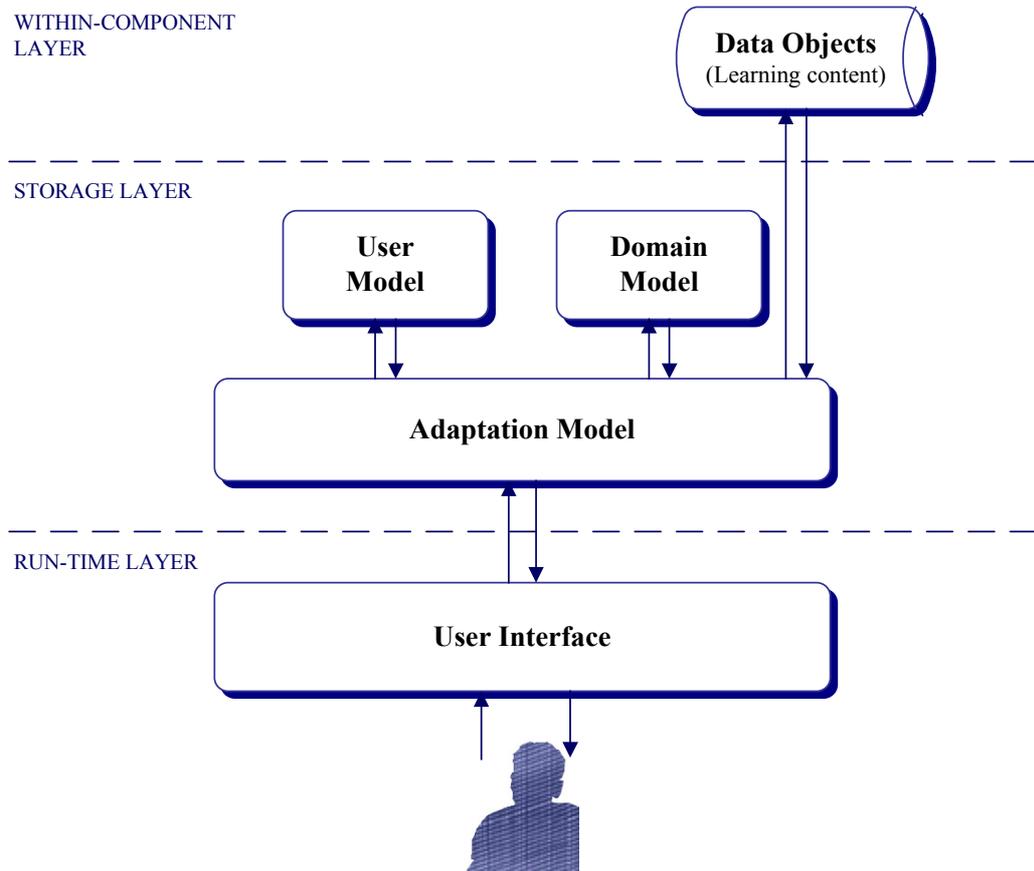
## **Adaptive Navigation Support**

Adaptive navigation support refers to the means that adaptive systems employ to help users to find their paths in hyperspace by adapting the visible links to which they have access. Techniques used include direct guidance and map adaptation, as well as sorting, hiding, annotating and generating links [Brusilovsky, 1996, 2001; De Bra et al., 2005].

- *Direct Guidance* – Users are directed to the next page of content. Unlike static systems with pre-defined sequences of content, adaptive systems may determine the next “best” page for the learner depending on their individual characteristics.
- *Map adaptation* – Navigation maps of websites can be more meaningful for the user if adapted to their individual needs; for example, by reducing the map size, or by annotating its nodes according to their relevance for the current task/content.
- *Link adaptation* – Adaptive systems may implement different strategies for sorting the links available in a webpage according to some user-valuable criteria; for example, showing the more relevant links closer to the top. Links can completely be removed from the list of links (by eliminating its anchor text), or just hidden to the user (by turning the link to the same colour as normal text). Annotations can also be used to indicate the relevance of the content behind the links, these annotations can be provided in textual form or using visual cues, such as icons, colours, font sizes, or font types. Links can also be adaptively created by matching or discovering relationships, such as the list of links that results from a search.

### **2.5.2 General Structure of AH Systems**

The Dexter Hypertext Model [Halasz & Schwartz, 1994] is widely used as a starting point for defining common abstractions of the functionality of hypermedia systems. Other models have been developed that capture specific approaches to the development of AH systems, such as the AHAM reference model [De Bra et al., 1999]; the Munich reference model [Koch & Wirsing, 2002]; or the Goldsmiths model [Ohene-Djan & Fernandes, 2002]. However, the Dexter framework represents a general approach that can be applied independently of the application’s domain or architecture. The conceptual structure that it proposes is organised into three layers: runtime layer, storage layer and within-component layer, as represented in Figure 2.3.



**Figure 2.3:** The conceptual structure of adaptive systems.

### **Runtime Layer**

It represents the user-interface. The Dexter framework does not prescribe what the user interface should do; instead it provides abstractions of how the content has to be delivered by means of presentation specifications.

### **Within-component Layer**

This layer contains the subject-specific data objects, which in the case of educational applications refer to the learning content of the system.

### **Storage Layer**

The Storage layer consists of three functionally different components: the domain model, the user model and the adaptation model.

#### *Domain Model*

The domain model provides a conceptual representation of the application domain, which may consist of concepts or tasks and the relationships between them, encoded in an appropriate representational formalism [Boyle, 1997; De Bra et al., 2005].

In educational AH systems, the domain model refers to the subject matter or curriculum being learned, particularly to the explicit representation of the knowledge an expert in the domain possesses [Boyle, 1997].

#### *User Model*

The user model is a conceptual representation of all the aspects of the user that are relevant for the adaptive application, such as an overlay model of the domain, or the user's background, experience or preferences [De Bra et al., 2005]. Although adaptation may be provided based on the information gathered during a single session, most adaptive applications gather information about their users over longer periods of time, for which a user model is stored permanently.

In educational adaptive applications, the term *learner model* is used instead. It keeps a record of the student's performance, which in combination with the learner preferences or individual characteristics is used to adapt the learning content [Boyle, 1997].

#### *Adaptation Model*

The adaptation model describes how the domain model and the user model are combined to generate adaptation; it generally consists of a set of adaptation rules that either define a user model update or a desired adaptation in the presentation of its content [De Bra et al., 2005].

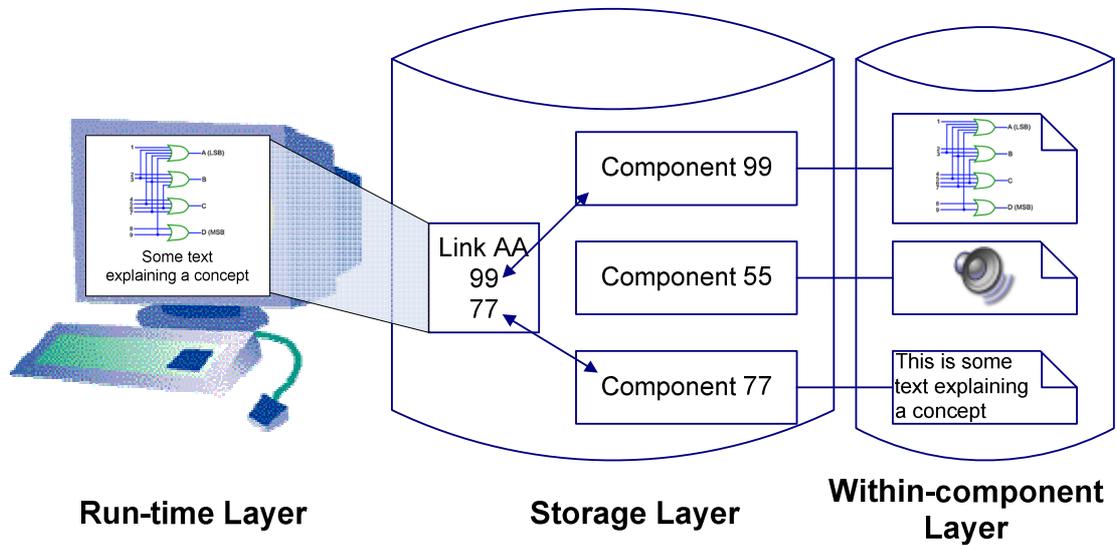
In educational AH systems, the adaptation model uses the information gathered about the learner and about the subject matter to plan the next teaching intervention.

### **2.5.3 Basic Functionality of Educational AH Applications**

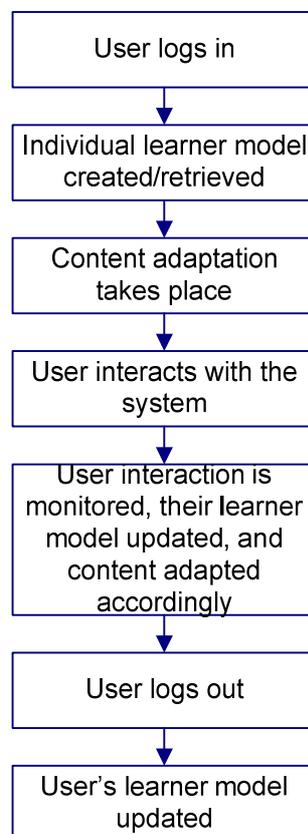
While different applications are based on different approaches and architectures and thus exhibit different behaviours, the structure proposed by the Dexter model is useful to acquire a general understanding about how educational AH applications work.

When a learner logs in an educational AH system, the system looks for his or her learning model, or creates a new one if it is the first time the student uses the system. Once the learner model has been retrieved, the adaptation engine identifies its current state and, taking into account the domain model, determines the most suitable content to be presented. When the appropriate learning content has been identified, the system

retrieves the actual files from the within-component layer and renders the pages of content that would be displayed through the user's interface.



**Figure 2.4:** An example of the three layers of the Dexter model as embedded in an educational AH system. [Adapted from Halasz & Schwartz, 1994, p. 33].



**Figure 2.5:** Simplified flow chart showing how AH systems work.

Figure 2.4 exemplifies the three layers of the Dexter reference model. In this figure, the storage layer contains the various components of the domain being learned, from these, the adaptation model has identified two components (*Component 99* and *Component 77*), which constitute the most suitable learning materials to be presented to the student given the current state of his/her learner model. The actual files of learning content (text, graphics, audio) are located in the within-component layer. In the run-time layer, a content page is presented to the user as a response to their request. The system keeps track of the students' actions and learning progress and uses this information to update their learner model, which in turn is used by the adaptation model to redirect the session accordingly. A simplified diagram of the whole process is presented in Figure 2.5.

## **2.6 AH APPLICATIONS IN EDUCATION**

A wide range of AH systems have been developed since the 1990's and it would be impractical to review all existing AH systems here. The discussion below is restricted to some examples that demonstrate the development of AH systems in education from early approaches, mainly focused on exploiting the technical capabilities of the Internet and the Web, to more robust systems integrating educational principles with technological facilities to provide personalised learning support. The review examines the selected applications considering their general functionality, their adaptive features, and their underlying structure in terms of the models described previously.

### **2.6.1 ELM-ART**

ELM-ART [Brusilovsky et al., 1996] was one of the first adaptive hypermedia applications developed for the Web. It was an online textbook with an integrated problem solving environment aimed at supporting learning programming in LISP.

ELM-ART was organised around three main components: reference manual, textbook and problem solving support. Like the index of a traditional book, the reference manual listed all the concepts contained in the textbook, when an entry was selected, an introduction to the concept was presented to the student as well as links to related units. The textbook was hierarchically structured into chapters, sections, subsections and units. Each unit was presented as a webpage including the description of its related concepts (text, images, and pieces of code) as well as further links to related units. The problem solving support feature provided examples and exercises selected according to

their relevance for the current concepts being learned and taking into account the students' previous experience –i.e. what concepts they had already studied.

Adaptive navigation support was provided by means of link annotation and link sorting. Links were annotated using different icons, fonts and colours to indicate when the content of a unit was “already known”, “ready to be learned” or “not ready to be learned”. Links to further concepts were sorted according to their relevance for the current concept and taking into account the concepts previously studied by the learner.

**Domain model.** ELM-ART's domain model was organised around a network of domain concepts and the relationships between them. The two main types of relationships in this structure were “part-of” and “is-a”, from which prerequisite relationships between concepts were computed using several heuristics.

**Learner model.** The learner model was an overlay of the domain model. A permanent model was maintained for each registered student in which their conceptual knowledge was recorded. Visual cues and the order of links were computed dynamically from the individual learner model.

**Adaptation model.** Content pages presented to each student were generated on the fly, based on the domain model and the individual learner model. The actual content of the course was stored as an annotated HyperText Markup Language (**HTML**) file. When a page was requested, the system extracted the content from the HTML file, and then links were determined from the domain model using various heuristics. Adaptive link annotation and link sorting were computed from the current state of the learner model.

ELM-ART was a very innovative application for its time. Using a textbook metaphor, structured learning materials were made available online, providing adaptive navigation support at the individual level. The domain model used in ELM-ART was at the centre of its adaptive functionality. A carefully crafted conceptual structure was used where all concepts relevant for the course were included and organised into chapters, sections, sub-sections and units, in accordance with the concept of a textbook. High-level relationships were used to connect the concepts between them, and heuristics applied at runtime to determine their prerequisite relations. Adaptation was provided at the individual level by maintaining a permanent learner model of each student. Data was implicitly collected recording new concepts studied by each learner, from which their learner model was continuously updated.

## 2.6.2 INTERBOOK

Based on the experience gained with ELM-ART, INTERBOOK [Brusilovsky et al., 1996b] sought to generalize the approach and to develop a subject-independent tool to simplify the process of creating adaptive electronic textbooks.

The online textbooks created using INTERBOOK were organised around two main components: a glossary, including all the concepts of the course; and the content of the textbook hierarchically structured into chapters, sections, sub-sections and units.

For authors using INTERBOOK the design process started with a list of all the domain concepts required. A Word document was required listing all these concepts, indicating by means of text styles the hierarchy of each entry – i.e. using “Header 1” text style for the titles of the chapters in the textbook, “Header 2” style for the sections, “Header 3” style for the sub-sections, and so forth. Each entry at the unit level had to be annotated indicating its prerequisite concepts and outcome concepts. The resultant file was then translated into HTML using a special tool provided by INTERBOOK. When the INTERBOOK server started, files were parsed and the structure of the textbook created.

The interface presented to the students consisted of two main windows; the first one presented the glossary with the list of concepts available in the course. When a concept was selected, its description and the list of corresponding units in the textbook were presented. The second window was then updated to show the content selected, links to related concepts and lists of prerequisite and outcome concepts.

Adaptive navigation was provided in terms of link annotation and prerequisite-based help. Links were annotated using visual cues to show the state of each concept: white icons in front of content “already known”, green for content “ready to be learned”, and red for content “not ready to be learned”. Prerequisite-based help referred to the capability of INTERBOOK to sort the links to related content according to the learner’s knowledge: more helpful concepts (those not known) were presented first.

**Domain model.** The domain model was structured as a network of domain concepts and their relationships. This network was a hierarchical structure organising the content into units of different level. Instead of using heuristics to determine relationships between concepts (as it was done in ELM-ART), INTERBOOK introduced the concept of “indexing”: for each unit, a list of related concepts was provided by the course designer, indicating the role of each concept either as an outcome or as a prerequisite.

The domain model was externalised in the form of a glossary, where each entry corresponded to one of the domain concepts. In turn, indexing provided the system with knowledge about the content of its pages, i.e. which concepts corresponded to each unit, which concepts had to be known already and which concepts would be known as a result of studying the current unit.

**Learner model.** The conceptual network created for the domain model also served as an overlay model for representing the user's knowledge about each concept, allowing the system to dynamically determine the visual cues and the order of the links.

**Adaptation model.** Content pages were created on the fly taking into account the domain model and the learner model. Since the system knew about dependencies between concepts in any unit, it was able to select and extract its content from the HTML file where the learning content was stored and to determine the list of prerequisite and outcome concepts. Adaptive annotation and sorting of links was computed based on the current state of the learner model.

INTERBOOK became one of the first tools for creating adaptive Web-base applications for learning. It seriously simplified the process of producing electronic textbooks, supporting both novice authors – who required using just MS Word, and more experienced programmers – who were able to make advanced changes to the HTML files produced. Moreover, INTERBOOK introduced the idea of indexing as the means to provide the system with knowledge about the domain: each entry at the unit level had to be annotated indicating its prerequisite and outcome concepts. Pages of content were created using the system's knowledge about the learning content and adaptation was provided by maintaining a permanent learner model. An implicit data collection process was used to register the new concepts studied by the learner and to update the user model.

### **2.6.3 AHA!: Adaptive Hypermedia Architecture**

AHA! [De Bra & Calvi, 1998; De Bra et al., 2005] is a generic development environment to create adaptive hypermedia software for different application areas – although educational content was its original purpose. The adaptive behaviour of the applications developed using AHA! is based on a conceptual model of the domain structured around a series of concepts and their relationships. In turn, this structure is

translated into low-level adaptation rules used by the system to update the model of the learner and to adaptively generate the content.

Adaptation observed by the learners includes the adaptive choice of link destinations, that is, when they select a link on their Web browser the system applies a series of rules to determine what page to show, which depends on the structure of the domain model and the current state of their learner model. Adaptation also includes the conditional inclusion of information, i.e. showing or hiding elements in the content pages depending on the learner model. AHA! also supports different adaptive navigation techniques.

For authors, AHA! is an open source software based on Java servlets running over a Tomcat Web server. As a development environment it offers a number of tools to define the concepts and concept relationships that determine the structure of an application. Other facilities include a graphic author tool that facilitates the visualisation of the domain; templates that support the editing of concepts and concept-relationship types; a form editor; a module for creating multiple-choice tests and a layout manager.

**Domain Model.** In AHA! the domain model is a concept hierarchy that consist of concepts and concept relationships. A concept is an abstract representation of an information unit in the subject domain. Concept relationships define the various types of interconnections between concepts.

Within the system, each concept consist of an unique identifier, a sequence of anchors that identify the substructure of the concept, and a set of attribute-value pairs that list the properties of the concept and the variables that would be used to generate adaptation during runtime. Concept relationships consist of a unique identifier, the sequence of the concepts it interconnects and a set of attribute-value pairs. Figure 2.6 presents an extract of a concept template used in AHA!; it is implemented in eXtensible Markup Language (XML) and includes a series of *<attribute, value>* pairs for an atomic concept.

**Learner Model.** The learner model is an overlay of the domain model structure; as such, it consists of a number of concepts with a series of attribute-value pairs. An additional concept is included in this structure called “personal” which is used to store all the attributes that describe the user, such as their preferences, login and password.

**Adaptation Model.** The adaptation model consists of a set of adaptation rules of the type event-condition-action. When a learner clicks on a link in a page served by AHA!, the system first identifies to which concept it points out, and then executes the rules included in the attributes of this concept. Each rule may update the value of one or more

attributes in one or more concepts of the learner model, which in turn may trigger additional rules in the attributes involved. When all the concerned rules are executed, AHA! determines which content was requested. Processing the request may require the inclusion or exclusion of content fragments and links.

```

<?xml version="1.0"?>
<!DOCTYPE template SYSTEM 'template.dtd'>
<template>
<name>page concept</name>
<attributes>
  <attribute>
    <name>access</name>
    <description>triggered by a page access</description>
    <default>>false</default>
    <type>bold</type>
    <isPersistent>>false</isPersistent>
    <isSystem>>true</isSystem>
    <isChangeable>>false</isChangeable>
  </attribute>
  <attribute>
    <name>knowledge</name>
    <description>knowledge about this concept</description>
    <default>0</default>
    <type>int</type>
    <isPersistent>>true</isPersistent>
    <isSystem>>false</isSystem>
    <isChangeable>>true</isChangeable>
  </attribute>
  <attribute>
    <name>visited</name>
    <description>has this page been visited?</description>
    <default>0</default>
    <type>int</type>
    <isPersistent>>true</isPersistent>
    <isSystem>>true</isSystem>
    <isChangeable>>false</isChangeable>
  </attribute>
  <attribute>
    <name>suitability</name>
    <description>the suitability of this page</description>
    <default>>true</default>
    <type>bold</type>
    <isPersistent>>false</isPersistent>
    <isSystem>>true</isSystem>
    <isChangeable>>false</isChangeable>
  </attribute>
</attributes>
<hasresource>>true</hasresource>
<concepttype>page</concepttype>
<conceptrelations>
  <conceptrelation>
    <name>knowledge_update</name>
    <label>35</label>
  </conceptrelation>
</conceptrelations>
</template>

```

**Figure 2.6:** Example of a concept template in AHA!  
 [Source: De Bra et al., 2005, p. 22.].

An example of the actual syntax of adaptation rules in AHA! is presented in Figure 2.7. Three rules are included which are conditionally executed when a component called *readme* of the application *tutorial* is accessed. The first rule is executed when *tutorial.readme.suitability* is false and *tutorial.readme.knowledge* is lower than 35, in which case the action carried out by the system is to assign a value of 35 to the *knowledge* attribute of the *tutorial.readme* component. The second rule assigns a value

of 100 to the *knowledge* attribute of *tutorial.readme* if the *suitability* of this component is true. The third rule states that the *visited* attribute of the *tutorial.readme* component would be set to 100 if its *suitability* attribute is *true*.

```

<generateListItem isPropagating = "true">
  <requirement> !tutorial.readme.suitability & &
    tutorial.readme.knowledge &lt; 35
</requirement>
<trueActions>
  <action>
    <conceptName> tutorial.readme </conceptName>
    <attributeName> knowledge </attributeName>
    <expression> 35 </expression>
  </action>
</trueActions>
</generateListItem>
<generateListItem isPropagating = "true">
  <requirement> tutorial.readme.suitability </requirement>
<trueActions>
  <action>
    <conceptName> tutorial.readme </conceptName>
    <attributeName> knowledge </attributeName>
    <expression> 100 </expression>
  </action>
</trueActions>
</generateListItem>
<generateListItem isPropagating = "true">
  <requirement> tutorial.readme.suitability </requirement>
<trueActions>
  <action>
    <conceptName> tutorial.readme </conceptName>
    <attributeName> visited </attributeName>
    <expression> 100 </expression>
  </action>
</trueActions>
</generateListItem>

```

**Figure 2.7:** Example of the syntax of adaptation rules in AHA!  
 [Source: De Bra et al., 2005, p. 32].

Since its earliest version, AHA! has sought to provide a wide range of adaptive techniques for different application areas. It concentrates its functionality on the provision of adaptive features leaving design and presentation issues to the authors – although support exists since version 3.0 in the form of a Layout Manager.

Adaptive behaviour supported by AHA! includes conditional fragments and various kinds of adaptive linking. Conditional fragments are included in the content files using special tags that exist in an AHA! extension to **XHTML** (eXtensible HTML). Adaptive linking is realised by means of link classes associated to different schemes.

One of the main characteristics of the AHA! framework is the arbitrary set of attribute-value pairs defined for each concept. While some specific types of concepts exist as templates in AHA!, authors can define additional types with different lists of attributes. Since the attributes and their values can be of any type, it is possible to define complex structures associating various sets of attributes for the purpose of managing adaptivity. As for the learner model, attributes can refer to any characteristic of the learner that is relevant for the application – knowledge level, behaviour, preferences, etc.

Creating adaptive applications requires careful design and planning, as well as the appropriate technical platform for its development. The facilities provided by AHA! for the development of adaptive applications have evolved as experience has been gained in the field. Currently, as open source software, different groups of researchers are contributing further to its development.

#### **2.6.4 TANGOW: Task-based Adaptive learner Guidance on the Web**

TANGOW [Carro et al., 1999] represented a different approach in educational AH applications because it regarded learning as a process more complex than navigating web pages and reading its content; also because it introduced the idea of structuring content into tasks and subtasks.

In TANGOW a course is structured into Teaching Tasks (TT). Teaching tasks are considered the basic unit of learning and may be atomic or composed of a series of subtasks. Each TT refers to theory, examples or exercises about the subject content, and has different media elements associated to it – text, images, video, sound, etc. The actual content presented at runtime depends on the state of the learner model.

When a student logs into TANGOW a new session is open and its individual learner model retrieved. Based on the information stored, a dynamic tree of achievable tasks and subtasks is created and the learner is presented with a list of links, starting with the last attempted task. Every time the student clicks on a link, the system updates the dynamic tree of available tasks. When a task is finished, the system verifies and records the execution results, which include the number of pages read, time spent on each page, number of exercises carried out and number of exercises successfully solved. This information is used to calculate a success value which is propagated up in the tree: if finalisation requirements of the parent task are satisfied and no further subtasks are available for the learner, the parent task ends. The system goes up the dynamic tree checking finalisation requirements of parent tasks until a task that has not been carried out is found. A new list of achievable tasks is then identified and presented to the user. When the student decides to finish the session, the system saves their execution results to reconstruct their personal dynamic tree in future sessions.

**Domain Model.** A repository of TTs is created using a database. Figure 2.8 shows the description of a teaching task, it consists of six fields: task type (theory, example,

practice, etc.), atomicity (atomic or composed), description, end method, parameters for the end method and list of media elements.

TYPE=	T
ATOMIC=	Y
DESCRIPTION=	Description of circular signs
END_METH=	F_TEO
PARAMS=	pag_visited tot_pag
HTML=	CIRCULARES STOP C_PROHI E_PROHI EP_VEHI EPV_SIDE

**Figure 2.8:** Description of a teaching task in TANGOW.  
[Source: Carro et al., 1999, p. 2.]

**Learner Model.** The learner model keeps a record of the learning tasks and execution results achieved by each student. This record is used to re-create the task tree at the beginning of each session.

Other information kept in the learner model relates to the student characteristics and preferences, including age, preferred language and preferred learning strategy. Media elements that appear in the content pages are selected depending on the student age and language; the learner’s age determines the level of difficulty of the tasks selected, while their preferred language is matched to the language of the media elements available. TANGOW also deals with two different learning strategies: “theory presentation first” and “practical exercises first”. Students can select their preferred strategy at the beginning of each session, which determines the sequence in which the system will present the subtasks to be carried out.

**Adaptation Model.** Adaptivity is based on a series of rules that describe how a TT is divided into subtasks, specify the relative position of the media elements in the content pages generated and state the activation conditions for the task. Activation conditions refer to prerequisite tasks, active learning strategy, and learners’ age and language.

Adaptive navigation is realised by applying the system rules to select the next set of achievable tasks, which are presented in the form of a menu, reducing in this way the navigation space available at any given moment of the session. In turn, content pages are generated on the fly based on the tree of achievable tasks and the learner model, from which the system determines what media elements to include and in what order.

Figure 2.9 presents an example of an adaptation rule in TANGOW; it shows the description of a rule indicating that the “Priority” task is composed of two subtasks: “Circumstantial\_Signs\_Theory” and “Circumstantial\_Signs\_Exercises” which have to be carried out in a fixed order (AND sequencing). The activation condition for this rule is called “c-4” for which the parameters are “exer\_ok” from “Vertical\_Signs”, meaning that for this task to be applicable, the current learner should have already carried out the exercises included in the “Vertical\_Signs” task. The final slot of the rule indicates how the performance of a learner would be calculated, which considers the time spent on each subtask, the number of pages read, the number of exercises attempted and the number of exercises carried out correctly.

SEQUENCING=	AND
LHS=	Priority
RHS=	Circumstantial_Signs_Theory Circumstantial_Signs_Exercises
ACT_CONDITION=	c-4
PARAM=	exer_ok Vertical_Signs
CALC_PARAMS=	time_in msum2 time_in Circumstantial_Signs_Theory time_in Circumstantial_Signs_Exercises tot_pag mdirect tot_pag Circumstantial_Signs_Theory exer_ok mdirect exer_ok Circumstantial_Signs_Exercises exer_done mdirect exer_done Circumstantial_Signs_Exercises tot_exer mdirect tot_exer Circumstantial_Signs_Exercises pag_visited mdirect pag_visited Circumstantial_Signs_Theory

**Figure 2.9:** Description of a task rule in TANGOW.  
[Source: Carro et al., 1999, p. 3.]

TANGOW introduced the idea of structuring the subject content into tasks and subtasks of different type, which allowed the system to record student progress using executing results to calculate a successful value for each task.

TANGOW was also one of the first systems to explicitly incorporate user characteristics (age) and their preferences (preferred language and learning strategy) into its user model. Accordingly, content assets are available in different levels of difficulty, different languages and using different media.

### 2.6.5 ARTHUR

ARTHUR [Gilbert & Han, 1999] is a Web-based instruction system aiming at delivering a many-to-one instructor/learner relationship. It takes different styles of instruction from a number of instructors and makes them available to its users. A group of instructors collaborate to create a map of the course content organised into concepts – basic units of

instruction that must be covered within the course. Each instructor then produces a module that adheres to this map.

Instructors submit their course modules via Web-based forms indicating the Uniform Resource Locator (**URL**) that points to the first webpage associated with each concept within the module – this allows instructors to maintain their course materials in their own Web servers. Instructors also submit quizzes to assess learning on each concept. When content is added to the system, each module is classified as auditory, visual, tactile or text-based.

Learners access ARTHUR using a pre-assigned username and password. The system then delivers the first concept of the course from the instruction pool available; the style of this initial module is selected at random. When learners complete a concept they have to answer the corresponding quiz. Quiz scores are then used to adapt instruction: if a learner scores 80% or above ARTHUR will move to the next concept using the current instruction style, if a score is lower than 80% the system will present the same concept using a different style.

**Domain Model.** ARTHUR uses a knowledge base stored in a database with different tables that organise information about learners, instructors and learner-instruction interactions. The domain model comprises a series of interrelated concepts as detailed by the instructors in the course map. A table is kept that includes these concepts, the URLs of their related modules and the instruction style of each module.

**Learner Model.** Information kept about the learners includes their general details such as name, login and password, as well as details of the concepts studied, the style of the course modules delivered and the results obtained in each concept quiz.

**Adaptation Model.** As part of the learner model a record is kept detailing the results obtained in each quiz. This information is used to classify learners and to adapt instruction. Using case-based reasoning new users are classified by matching previously observed cases; if a new learner exhibits similar results to a previous learner, the system assumes that both students have similar styles and assigns to the new learner the instruction style previously followed by the older student.

Based on theory indicating that people exhibit different approaches to learning and studying, ARTHUR facilitates access to learning material developed by a number of instructors and possibly hosted in different Web servers. Instructors have to follow the

same course structure and deliver the same concepts using different instructional styles by means of different media elements of auditory, visual or textual type. Learners are then classified and presented with course modules of a given style. Data is implicitly collected recording assessment results that are used in case-based reasoning to adapt the instruction style of the next concept.

ARTHUR implemented a series of technologies to provide access to distributed learning materials through a consistent user interface, including applets, dynamic HTML and audio and video streaming. Its knowledge base is organised around a database with multiple tables that provide the basic facts to apply case-based reasoning. While learners are assigned to specific stereotypes, these are not permanent and different instruction styles are delivered based on the assessment results of each concept. One possible drawback is that ARTHUR completely depends on the instructors to develop learning content that truly adhered to a particular style (visual, auditory or textual).

#### **2.6.6 AES-CS: Adaptive Educational System based on Cognitive Styles**

AES-CS [Triantafillou et al., 2003] is an educational AH system developed to provide different instructional modes to students with different cognitive styles. It uses Witkin's [1978] model of cognitive styles to classify students as field-dependent (FD) or field-independent (FI).

Based on the style of the learner, AES-CS adapts a series of elements in the learning environment, including approach to the learning content, level of learner control, navigational aids and amount of instructions and feedback. FD learners are presented with a global approach to the learning content, and FI learners with an analytical approach. For this purpose, AES-CS uses various adaptive techniques including conditional text and page variants.

AES-CS provides both program and learner control. In the case of learner control, the system provides a menu from which students can progress through the course in any order. In the program control option the system guides the user through the learning material via direct guidance.

Adaptive navigational aids include the provision of advance organisers for FD learners and post organisers for FI students. FD learners receive further support through the availability of a concept map and a graphic path indicator.

Since FD learners are more likely to require extensive instructions and feedback, an additional frame is used to provide extensive guidance for them. Conversely, instructions and feedback are kept to a minimum in the case of FI learners.

**Domain Model.** Content is organised around a set concepts subdivided into topics that represent the basic units of knowledge for the domain. Topics are related to each other thus forming a semantic network which is in fact the structure of the domain. In the system each hypermedia page corresponds to one topic only.

**Learner Model.** An overlay of the domain structure is implemented in the learner model to record the users' progress through the course content – indicating whether a concept is unknown, known, learned or well-learned. In addition, a series of variables are included as a cognitive profile, indicating the learners' cognitive style and their preferences in terms of learner control, navigational aids and amount of instructions.

**Adaptation Model.** Based on the current state of the learner model and the model of the course domain, AES-CS applies a number of adaptive techniques to create the content pages and to provide navigation support. Conditional text is used to determine whether an information chunk is presented to the learner or not. Also, following the page variants technique, two alternative pages associated with each topic are prepared; each variant presents information in a different style according to the FD-FI dimension.

Adaptive navigation support is provided in the form of link annotation and direct guidance. Link annotation is used to indicate the student when a link is “recommended” (blue), or “not ready to be learned” (grey). The selection and presentation of links is adapted taking into account the learner's knowledge state, style and preferences. In the case of direct guidance, the system leads the learner to the next page of content.

AES-CS was one of the first systems to explore the implications of using cognitive styles in the design and development of adaptive educational applications. Based on Witkin's model of cognitive styles, a series of variables are used to adapt the learning content and suit individuals of field-dependent (FD) and field-independent style (FI). The initial configuration of the system provides FD learners with program control, structured lessons and maximum instructions and feedback. For FI learners the initial configuration provides learner control, minimum instructions and feedback, and allows students to follow different paths through the learning content. Implicit data collection is implemented to record the concepts studied and to update the learner model

accordingly. While the cognitive style of the learner initially determines the configuration of the system, AES-CS allows their users to inspect and modify some variables in their learner model, including control strategy and amount of instructions.

### **2.6.7 INSPIRE: INtelligent System for Personalised Instruction in a Remote Environment**

INSPIRE [Papanikolaou et al., 2003] strongly emphasises the integration of instructional design theories in the development of adaptive educational applications. INSPIRE integrates ideas from “Elaboration theory” for selecting, sequencing, synthesising and summarising content, and from "Component display theory" for providing the appropriate type of learning content at different levels of performance. These principles are combined with learning styles theory to provide adaptivity.

INSPIRE supports their users by restricting the domain knowledge to which they have access and enriching it progressively according to their performance. When learners access the system they select a topic from the course domain and INSPIRE generates lessons based on their knowledge level, previous performance and learning style. The lesson content appears as a list of hyperlinks in a menu. Initially, the menu includes links to introductory concepts and, as the learner progresses through the course, more complex concepts are added. Further support is provided by means of adaptive content presentation and adaptive navigation based on the learners’ performance and style.

**Domain Model.** The domain model is organised into three hierarchical levels of knowledge abstraction: topics, concepts and learning materials. Each topic corresponds to a learning goal of the course; in turn, each topic consists of a set of concepts that are classified as prerequisite or outcome concepts. Outcome concepts are organised further into a hierarchical structure: introductory concepts are found at the highest level while subsequent levels include higher complexity or detail.

The learning materials developed for each concept include various external representations, such as theory, examples, exercises, activities and assessment tasks. Learning materials support learners to achieve three levels of performance: “remember” (introduce the concept), “use” (apply the concept) and “find” (find new applications for the concept). Questions included in the assessment test of each concept are classified using the same categories.

**Learner Model.** Information about learners is stored using a hierarchical structure: their general profile is kept at the highest level, including username, password and learning

style. The second layer stores information about the topics studied, including the time spent on each topic and the name of the current one. Information on the third layer refers to the concepts studied within each topic. The last layer stores data about the knowledge level achieved by each user after studying the concepts selected.

The learner's knowledge level on each concept is estimated using their results on the assessment tasks. Considering the different categories of questions included in the assessment tests (i.e. "remember", "use" and "find"), as well as the type of learning content studied (i.e. theory, examples, exercises, etc.), the system applies a complex diagnosis algorithm that determines whether the knowledge level of the learner is "inadequate", "mediocre", "advanced" or "proficient".

**Adaptation Model.** INSPIRE adaptive behaviour involves curriculum sequencing, adaptive navigation support and adaptive content presentation. The learners' knowledge level and their learning styles are used as the main sources of adaptation.

In terms of curriculum sequencing, concepts within each topic are presented gradually following their layered structure. Initially, just the basic concepts are included and to present the next layer (i.e. concepts with higher complexity or detail) the learner should be "advanced" on the concepts of the current layer. The learner's knowledge level also determines the type of learning materials recommended by the system.

Adaptive navigation support is implemented by means of link annotation. Visual cues are used to indicate whether a link in the menu is recommended or not, to highlight the type of the learning materials behind a link, and to indicate when a link has been visited.

Alternative presentations of the educational materials are used depending on the student's learning style. While the same materials are delivered to all learners, the order in which these are presented differ; for example, if the style of the learner is "reflector" then learning materials of the type "example" would appear at the top of the content page followed by links pointing to theory presentations and exercises.

INSPIRE implemented a different approach combining a fairly complex domain model with assessment of learners' performance and curriculum sequencing. A domain model structure is proposed around different levels of abstraction, with learning materials supporting different levels of performance and comprising different types of learning content to structure lessons for students with different learning styles. Considering the performance of the learner on different categories of questions, estimations are made

about their knowledge level, which is then used for planning the sequence of the content.

This framework clearly highlights several processes that affect the adaptive behaviour of educational AH systems, such as structuring the domain knowledge; planning the content, delivery, presentation and development of learning materials; assessing learners' knowledge; determining the appropriate sequencing of the learning content; adaptively delivering the educational material; and providing the appropriate navigation support. INSPIRE also draws attention to the relevance of developing modular, re-usable learning content. It also stresses the importance of giving the learner access to their learner model and control over the level of adaptation of the system.

## **2.7 PERSONALISING WEB-BASED LEARNING**

In previous sections it has been argued that LT is seen as a viable option for improving attention to diversity and for enhancing the individual learning experience. In particular, the use of the Internet and the Web for the provision of learning materials seems to be suitable for different learning settings and requirements. In addition, the capabilities of these technologies can be used to support the learner at an individual level unavailable to other artefacts. This in turn is in line with current theories that advocate the idea of learning as the individual process of constructing knowledge and teaching as the process of supporting that construction.

Towards this approach various classifications for the personalisation of Web-based content and services have been suggested. These are based on factors such as input data, collection methods, system predictions and user involvement, among others [Bental et al., 2001; Cranor, 2003]. Adaptive Hypermedia, a research approach that integrates knowledge from hypermedia systems development and user modelling, provides a framework for realising personalisation. AH systems incorporate features of adaptable and adaptive nature to modify their functionality to better match the user's requirements and characteristics. For this purpose, various techniques have been proposed to provide individual support at the navigation and content levels.

The generic structure of AH systems comprises a series of models that interact to determine their adaptive behaviour, namely domain model, learner model, and adaptation model. In educational AH systems the learner model represents these characteristics of the learner that are relevant for the system to deliver personalised

content, the domain model details the structure used to organise the subject content, and the adaptation model describes how the domain and the user model are combined to provide adaptive functionality.

The educational AH systems reviewed in the previous section exemplify how different approaches to personalisation have been implemented during the past years. Adaptive navigation support in the form of link annotation, for example, is provided in several AH systems such as ELM-ART, INTERBOOK, INSPIRE and AHA!, where available links are annotated as recommended, non recommended or neutral. Link sorting is available in applications such as ELM-ART and INTERBOOK, where most useful links are placed on top of the list. Link hiding and link removal are techniques available in AHA!.

Adaptive curriculum sequencing has been implemented in TANGOW and INSPIRE where the system determines the most suitable sequence of learning tasks to work with, thereby reducing the navigation space. Adaptive presentation has been realised by means of conditional fragments as in AHA!, or by adaptively selecting and sorting media elements to assemble content pages as in TANGOW, INSPIRE and AES-Cs.

Different approaches also exist for structuring the learning content. Early systems, such as ELM-ART, were based on structured networks of concepts from which prerequisite relationships were derived at runtime to determine the appropriate annotation of links. Later on INTERBOOK introduced the concept of “indexing”, whereby content designers had to explicitly indicate prerequisite and outcome relationships between concepts. Also, based on structured concepts and explicit relationships between them, AHA!, AES-Cs and INSPIRE use hierarchies to formalise their domain models. A different approach is implemented in TANGOW, where learning content is structured into teaching tasks.

Adaptation decisions are determined by the rules embedded in the adaptation model taking into account the structure of the learning content and the current state of the learner model. Usually, an individual model is maintained for each student, which details their knowledge about concepts of the domain (ELM-ART, INTERBOOK, AHA!, INSPIRE and AES-Cs). Other user characteristics have been used such as learner’s age and preferred language, learning strategy, learning style or cognitive style as in TANGOW, INSPIRE and AES-Cs.

Learners’ observable behaviour is in many cases the basis for determining their learning progress. In systems such as ELM-ART knowledge is generated by the student reading a content page. In other systems, such as INTERBOOK, AHA! and AES-Cs, concepts studied

by the learner and the prerequisite relationships between these concepts are used to update the learner model. A different approach is implemented in TANGOW, where task execution results are used to calculate a success value that determines whether prerequisite tasks have been successfully carried out. Similarly, INSPIRE implements a complex diagnosis algorithm that incorporates ideas from the fields of fuzzy logic and multi-criteria decision making to estimate the knowledge level of its learners.

Table 2.2 compares the educational AH systems reviewed in terms of the adaptivity provided and the structure of their domain and learner models.

From the analysis previously carried out it is clear to observe that the three models supporting AH systems work in a very coordinated way to provide adaptive functionality. A clear tendency is also observed through the examples reviewed: from a strong emphasis on the design of the domain model in early applications, to a design approach based on the learner characteristics in more recent educational AH systems. This seems to suggest that educational AH systems are increasingly incorporating a learner-centred approach, which recognises that if knowledge is individually constructed, the instructional approach to scaffolding the learning process has to primarily consider the learning needs of students. Therefore, attention to the user model is critical.

Brusilovsky [2001] points to two aspects that still challenge research into user modelling in AH: user interests, and individual differences. In terms of user interests, the rise of Web-based information retrieval has produced a considerable body of research on various aspects of personalising the information presented to the user. These approaches try to model users' long-term interests to use them in combination with short-term goals in order to improve information filtering and the system performance in general [Brusilovsky, 2001; De la Flor, 2005].

With regard to individual differences, the extent of the research carried out to date is more limited. While AH researchers agree on the importance of modelling and using individual attributes, and some of them have already begun exploring their use for personalisation (e.g. INSPIRE and AES-CS), it is still not clear which aspects are worth modelling and how to use them to treat different users differently. Brusilovsky [2001]

AH system	Subject Content	Domain Model Structure	Learner Model Structure	Adaptivity
ELM-ART [Brusilovsky et al., 1996]	Programming in LISP.	A network of concepts and their prerequisite relationships.	Overlay of the domain model. Individual record of concepts learned.	Adaptive link annotation and link sorting based on concepts studied by the learner. Problem solving support.
INTERBOOK [Brusilovsky et al., 1996b]	Subject-independent tool for creating adaptive electronic textbooks.	A network of indexed concepts indicating their prerequisite and outcome relationships.	Overlay of the domain model. Individual record of concepts learned.	Adaptive link annotation and prerequisite-based help (link sorting), based on concepts studied by the learner.
AHA! [De Bra & Calvi, 1998; De Bra et al., 2005]	Generic development environment.	Hierarchical structure of concept and sub-concepts. Every concept includes a set attribute-value pairs that are translated into adaptation rules.	Overlay of the domain model plus additional concepts that store the characteristics of the learner that are relevant for the system to provide adaptivity.	Adaptive navigation support: link annotation, based on concepts studied by the learner and realised by defining classes of links. Adaptive content presentation: conditional fragments can be indicated by the authors in their content files.
TANGOW [Carro et al., 1999]	Vehicle driving course.	Learning content structured into teaching tasks and subtasks. Structure details the type of task (theory, examples, and practice), whether it is atomic or composed, a method and associated parameters to finalise the task, and associated media elements.	Individual record of tasks carried out and execution results. User characteristics and preferences: age, preferred language, active learning strategy.	Adaptivity based on a series of rules and the use of a dynamic tree of active tasks. Adaptive navigation realised by applying rules to select a sub-set of teaching task, reducing the navigation space. Adaptive content based on current tasks and user characteristics.
ARTHUR [Gilbert & Han, 1999]	Physics 101.	A course map organised around basic concepts.	Record of concepts studied, their instruction style and quiz results.	Instruction style determined through case-based reasoning according to the information stored in the learner model.
AES-CS [Triantafillou et al., 2003]	Multimedia Technology Systems	A network of interrelated concepts and their constituent topics.	Overlay of the domain model. Individual record of concepts learned. User characteristics and preferences: cognitive style, control strategy, navigational aids and amount of instructions.	Adaptive content presentation: conditional text and page variants to present learning content according to the style of the learner. Adaptive navigation support: link annotation and direct guidance.
INSPIRE [Papanikolaou et al., 2003]	Computer Architecture	A complex hierarchical structure of learning topics consisting of prerequisite and outcome concepts of different level of difficulty. In turn, concepts are related to a series of modules of learning content.	Overlay of the domain model recording progress through topics and concepts. User characteristics, learning style and level of performance are also stored.	Curriculum sequencing, adaptive navigation support and adaptive content presentation, determined using the learner's knowledge level and learning style as main sources.

**Table 2.2:** Comparison of educational AH systems.

suggests that in order to make progress in this particular area, further work is required exploring the relationship between individual differences and possible interface settings. There is therefore a need for further research regarding individual differences and interface and interaction design which could make possible the advancement of educational AH systems towards the level of personalisation required by current approaches to learning and teaching such as constructivism.

Individual differences are found in the users' physical capabilities, intellectual skills, gender, personality, affective traits, cognitive characteristics, experience, social and cultural background, motivation and goals [Jonassen & Grabowski, 1993]. Some questions can be asked about the differences between individuals: what differences have a real impact on the human-computer interaction? Which of these differences are stable and useful? What is their impact when the interaction is for learning?

One of the main problems associated with individual differences is that some of them are so complex that describing them in terms of a manageable number of discriminative characteristics is practically unfeasible [Jonassen & Grabowski, 1993; Riding & Rayner, 1998]. Intelligence, for example, comprises such a considerable range of intellectual abilities that cannot be easily correlated with specific learning requirements. Personality refers to a complex mental disposition to interact with the environment and other people which hardly can inform the design of learning settings. Some other characteristics such as motivation, background and experience vary for the individual when taking into account factors such as subject matter or time available [Cristea & De Bra, 2002]. Cognitive characteristics, on the other hand, describe how individuals interact with their environment, extract information from it and construct and organise personal knowledge [Jonassen & Grabowski, 1993; Riding & Rayner, 1998]. While cognitive controls are closely related to mental abilities and their individual variations are so numerous that it makes the task of describing such variations ineffective, cognitive styles describe more general perceptual and processing characteristics whereby variation can be explained in terms of the combination of a relatively small number of dimensions, then the task of identifying these and using them for the design of instructional approaches is possible and potentially useful [Jonassen & Grabowski, 1993; Riding & Rayner, 1998]. More importantly: cognitive styles are considered to be a stable approach of each learner to knowledge acquisition [Benyon, 1993; Riding & Rayner, 1998; Atkinson, 2001; Webster, 2001; Cristea and De Bra, 2002]. If cognitive

styles are individual and non-changing, they could provide significant basis for modelling learners towards personalisation [Uruchurtu et al., 2005].

The question is then: is it worthwhile to take into account cognitive styles for the adaptation and personalisation of the computer interface for Web-based learning materials? It seems sensible to start answering this question by analysing the concept of cognitive styles, which is the aim of the next Chapter.

## **2.8 SUMMARY**

This Chapter has identified some of the driving issues and educational approaches behind personalisation of learning, which corresponds to objective one (O1) of the research. An appraisal of some influential pedagogical and technological trends has been carried out that illustrates how modern learning technology is increasingly moving towards a learning-centred approach and personalisation of learning. Learning technologies in general and Web-based systems in particular, have been characterised as viable, cost-effective alternatives for improving attention to diversity and enhancing individual learning experience.

An overview of the published literature relating to the different methods for dealing with individual differences of the users has been presented. Approaches to adaptivity and personalisation have been examined and selected adaptive hypermedia systems critically reviewed. These aspects are relevant for the research as stated in objective two (O2).

The need for further research regarding individual differences and interface settings has been identified. Cognitive styles of individual learners have been highlighted as a useful source to advance the development of educational AH systems towards the level of personalisation required by current approaches to learning and teaching. This is in line with objective three (O3) of the research.

In the next Chapter, cognitive styles are examined further and key characteristics that relate to the design of adaptive interfaces for Web-based learning materials are identified.