Panel Discussion:

Adaptive, inclusive software design for People with Vision and Hearing Impairments

Adaptive, inclusive software design is an area of research that aims to devise technologies and techniques to give equal access to educational materials and to assist all learners, regardless of impairment, in reaching there full educational potential within inclusive educational environments. An inclusive software design approach seeks to address the user interface and interaction needs of all learners with a specific focus on those who are vulnerable to marginalisation and exclusion. In educational environments inclusive software design is therefore concerned with providing appropriate responses to a broad spectrum of user interaction acquisition needs in formal and non-formal settings.

When designing inclusive software technologies for students in educational environments it is necessary to tailor both the use interface and educational material so that the learning needs of different students can be addressed. Adaptive, user interaction research investigates techniques used to extend traditional user systems with features, which enable a user's interaction to be personalized and content to be tailored on an individual basis. Such systems use information provided by, or captured from, learners to tailor the user interaction process. Through the application of this information, adaptive techniques can be used to support users in learning tasks by tailoring content and presentation, customizing user interfaces and providing additional guidance and support. Adaptive techniques have proved useful in learning tasks where users have differing learning requirements, disabilities, histories and preferences.

At present the majority of software learning technologies assume students have no impairments. When these technologies are used in classrooms there is the risk that certain students maybe excluded from learning tasks due to their inability to access or use the software. Furthermore, students with vision or hearing impairments are often marginalized as they have to use different technologies within the same classroom to accomplish the same tasks as those without disabilities.

This Panel will discuss the potential issues concerned with using adaptive, inclusive software design techniques to develop Software technologies for learners with impairments. Providing equality and parity in access to software-based educational materials in classroom environments where students have a range of impairments is therefore a serious challenge to designers of software is an important educational challenge.

Panel Discussion Questions:

1. How to establish a set of principles for adaptive, inclusive software design for People with Vision and Hearing Impairments
2. Formally modeling impairment possibilities and alternative interaction stages as user/learner models
3. Static or dynamic user/learner models of impairment?
4. Appropriate adaptive actions as responses to different impairments.
5. Architectural and design issues associated with adaptive, inclusive software design for People with Vision and Hearing Impairments

Background

It is well recognised that learning technologies when used to supplement traditional teaching methods can have a positive impact on learners. Learning technologies provide opportunities for learners to implement their knowledge in an interactive manner, increase motivations to learn and take pride in technical skills gained and personal development.

The use of adaptive techniques to personalise learning technologies is also well recognised. Broadly, such systems adapt the content and presentation of the educational learning materials on the basis of a decision theory, implemented as an adaptive function that consists of a model that can be used to describe different types of learner and a set of rules describing what decision to take when a learner has a particular model.

Historically, models used to describe learners are based on the knowledge a learner possesses or has acquired about the learning materials within the system. This knowledge may be captured by the system before the user begins a session (static model) using the software or whilst the user in using the system (dynamic model). Knowledge the system amasses about a user is used to form and subsequently update a learner's model. The second component of the adaptive function is a set of rules that describe different adaptive actions that can be taken by the system. These rules are parameterised by a learner's model to deduce what adaptive action the system should take on behalf of the learner. An adaptive function therefore implements an inference engine that tailors content and presentation on an individual learner basis.

When taking an Adaptive, inclusive software design approach the designer implements the adaptive function using "a model of inclusion" expressed as a set of rules to tailor both the use interface and educational material so that the learning needs of different students can be addressed. These rules maybe understood as expressing the most appropriate form of adaptation of the content and presentation based upon an individual's user's impairments.

A second important difference between the work presented here at that of previous adaptive learning systems research is that where previously adaptivity was motivated by the need to tailor and manage the learning materials, here we are interested in inclusive access to the tutorial software system We believe once this is achieved, it is the responsibility of the author of the educational learning materials to determine the tailoring of the content and presentation of these materials for different users in different learning contexts and that users with impairments should not be treated as a special case of learner in this regard.

An example of Adaptive, inclusive software design Approach

The Maths Adaptive Tutorial System for disabilities (MATSD) is an advanced learning technology designed to tutor learners with or without disabilities in mathematics. Through the use of adaptive techniques and a dynamic user interface MATSD gives learners with hearing impairments, learning difficulties, motor control disabilities and those with no recognised disability, equal access to a learning technology and its embedded educational materials. MATSD demonstrates how adaptive techniques can be used to address the learning needs of various types of users within the same software system. The architecture of MATSD is shown in Figure 1. The MATSD interface interacts with a module for capturing disability specifications that describe a learners accessibility needs and a module to read translated question specifications. Translated question specifications are rendered and presented to learners as a tailored user interface.

At the centre of MATSD is an adaptive function that implements a decision-making algorithm. The adaptive function reads in a learner's model of disability and then uses this information to parameterise a set of rules that control tailoring actions over question specifications. Questions and associated instructions within MATSD are composed from a combination of text, graphics, audio content, Makaton symbolic representations, and formatting. These media elements are used by the adaptive function to generate translated question specifications. These describe which media elements are required and how these should be combined to maximise accessibility to educational learning materials.

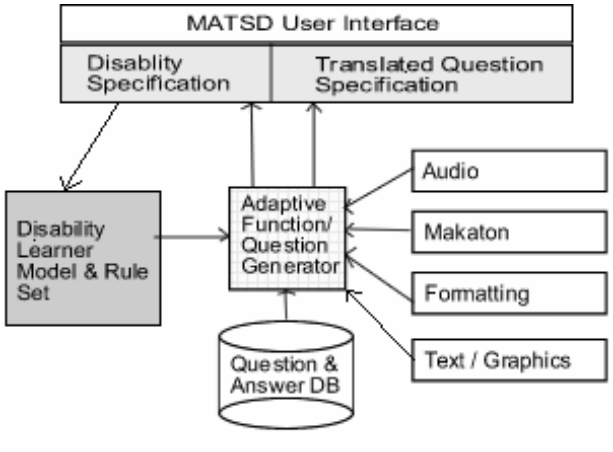

Figure 1. Architecture of MATSD

Within MATSD adaptive actions support inclusive education by providing translations of educational learning materials into symbolic representations using Makaton vocabulary, and audio translations.

Adaptive	**Learning Difficulties**	**Visual Impair-ments**	**Motor Control**
Text	Translated to Symbolic representation using Makaton	Resize and Audio translation	Let unchanged
Graphics	Symbolic representation using Makaton	Size increase Audio Descriptions	Left unchanged
Navigation	Visual Navigation Clues	Audio Navigation Trails	Large single click navigation
Questions	One Question per page Symbolic representation using Makaton	Size increase Audio Descriptions of Question	Spacing between question parts and user selections

Figure 2 Adaptive Actions of MATSD