Reusable Quality of Experience aware Adaptation Strategies for Authoring Adaptive e-Learning

Andreea Molnar, Cristina Hava-Muntean
School of Computing, National College of Ireland
Mayor Street, Dublin, Ireland
amolnar@student.ncirl.ie, cmuntean@ncirl.ie

Alexandra I. Cristea
Department of Computer Science, University of Warwick
Coventry, CV4 7AL, United Kingdom
A.I.Cristea@warwick.ac.uk

Abstract: With the increasing usage of multimedia in education, videos in particular, the delivery of content becomes difficult on low bandwidth networks. Conversely, the authoring of adaptable courses is still very difficult and time-consuming and no adaptation strategies for multimedia transmission customised for learner have been proposed so far. This paper presents two Quality of Experience aware reusable adaptable strategies for authoring adaptive and personalised content. These strategies are used by an author to specify adaptation rules in order to improve the quality of the educational material delivered to the learner. They are appropriate for any personalised learning environment that also caters for optimal media transmission.

Introduction

The usage of multimedia in education is continuously increasing. The delivery of such media is supported, due to regional variations in the developments in the area of communication technology, by different types of networks. This leads to learners who access networks with dissimilar performance characteristics. As a result, the educational content multimedia objects in particular (which constitute the currently most resource-intensive type of content to be transmitted so far) may have their integrity affected by poor network performance (low bandwidth, loss, delay, etc). This leads to a digital division between learners that have access to a high speed broadband link and those who do not.

The fact that poor quality of the multimedia affects learner satisfaction has been signalled in previous literature. For example, the subjects of two pilot studies conducted within the framework of the MUIS project, in 2005-2007, reported dissatisfaction due to multimedia quality (Milrad & Spikol 2007). Hooper et al. (2007) reported that by changing the multimedia frame rate (as a result also the multimedia quality), the subjects which participated in his study had they comprehension affected. Thus there is a need for a solution that addresses the efficient transmission of the multimedia to the learner and how to improve the Quality of Experience (QoE) the learner perceives. Muntean (2008) has proposed a framework that investigates the network performances and suggests content personalisation techniques that assures that the e-learner has access to the best possible educational content in terms of quality. Unfortunately, the process of authoring educational content and adaptation rules, while taking into account also the network parameters is at least time consuming. An authoring framework that addresses this issue has been proposed: Quality of Experience-LAOS (QoE-LAOS) and presented in Muntean et al. (2007).

The scope of this paper is to focus on the *QoE Rules sublayer*, a component of the QoE-LAOS that describes adaptation rules that enhance learner Quality of Experience. We are proposing two adaptation strategies that are used by QoE-MOT (Molnar & Muntean 2009) authoring tool in order to deliver educational content suitable to both learner profile and the learner network performance. Both strategies may be re-used by other authoring tools that follow the LAOS (Cristea & de Mooij 2003) model and use LAG (Cristea & Verschoor 2004) as a specification language for writing the adaptation rules.

The rest of this paper is organised as following. First, a literature review is presented. Then, a brief introduction on the Quality of Experience term, as used in this paper, follows. Then the QoE-LAOS authoring model is shortly sketched, with focus on the *QoE Rules sublayer* component. This is followed by the presentation of the proposed strategies and a discussion on their usage, advantages, as well as limitations and ways to improve them. The last section of this paper presents the conclusions we have reached so far and plans for future work.

Research Background

Delivering online content to *anybody*, *anytime* and *anywhere* is a challenging process. Among these challenges one can note offering personalised and adaptive content and the differences which exists in learner's network capabilities. Offering personalised and adaptive courses has been intensively studied in the last decade. Different aspects in personalisation have been considered such as: learner knowledge (Yudelson et al. 2008), goal (Karampiperis & Sampson 2005), prerequisites and experience (De Bra et al. 2006), emotional and physical state (Lukasenko & Grundspenkis 2009), etc.

One drawback of the current adaptive e-learning systems is that they are difficult to author, in particular the adaptation process (Brusilovsky 2003). In order to have personalised and adaptive courses, the authors need to spend far more time than when constructing systems which do not follow the personalisation path (Brusilovsky 2001). Different solutions have been proposed to make personalisation and adaptation more manageable, such as authoring tools and adaptation languages. They allow authors to create learning objects, define adaptation rules, etc. For example, MOT – My Online Teacher (Cristea & de Mooij 2003) allows learners to define content divided in concepts, assign attributes which describe them, change the order in which the concepts will be shown, and assign labels and weights which can be used during the adaptation process. The adaptation process can be specified through adaptation rules and various strategies. MEAT – Mobile E-learning Authoring Tool (Kuo & Huang 2009) provides authoring of multimodal interfaces. It produces SCORM (Sharable Content Object Reference Model) conformable learning objects that can be re-used by other e-learning systems. Wang et al. (2007) propose an authoring tool which integrates various e-learning standards (SCORM, IEEE LOM –IEEE Learning Objects Metadata, TW LOM –TaiWanese LOM), allowing the instructors to create compatible courses, but does not provide support for writing adaptation strategies. OpenMath (Manzoor et al. 2006) allows authoring mathematical content through an XML content editor. For facilitating the reuse of the material other systems were also proposed (Aroyo et al. 2002, Cristea & de Mooij 2003).

All the above mentioned authoring solutions do not take into account the fact that the educational content will not always be transmitted over networks with the same characteristics. Most educational systems are created under the wrong assumption that all learners have access to high speed technologies (Johnson et al., 2007). The solution we propose involves extending the authoring tool with new adaptation strategies that are learner network performance aware. The goal is to transmit to the learner the same educational content, but with a quality or in different forms, such that it would arrive to the learner in time and at an acceptable quality (the best possible based on his network). This paper focuses mainly on proposing reusable adaptive strategies and on how performance-aware adaptation may be done. The end scope is to allow the authors to easily create courses, accessible to *anybody, anywhere, anytime* which will also increase the learner Quality of Experience, when using the e-learning system.

Quality of Experience aware LAOS (QoE-LAOS)

Quality of Experience (QoE) is becoming more and more important, with the development of better communication technologies. Its importance is well illustrated by Kumar (2005) which says: "The consumer is king – and needs high QoE". Some researchers consider that this increasing importance comes from a more consumer-oriented mentality (Kumar 2005) and user-centred designed products (Sleeswijk et al. 2007). QoE may be affected by the network parameters. Poor conditions during the delivery will result in poor end user quality. *The perception of the network performance relative to the user expectation is referred to as Quality of Experience* (Odence 2004).

Little work was done on assessing QoE with e-learning systems. A QoE aware adaptive learning system, which assesses the learner QoE and provides personalised content, was proposed by Muntean (2008). In order to ease the creation of educational content, a *Quality of Experience authoring model* has been proposed: QoE-LAOS (Muntean et al. 2007). It aims at helping the authors to develop performance aware content selection mechanisms. The authoring model is based on LAOS (Cristea & Mooij 2003), a more generic authoring model for adaptive hypermedia. LAOS is a five layer model containing *Domain Model* (DM), *Goal and Constraints Model* (GM), *User Model* (UM), *Presentation Model* (PM) and *Adaptation Model* (AM). This model provides us with the necessary structure for the creation model of personalised learning content. In order to apply however QoE characteristics, the QoE-LAOS model adds three sublayers: the *QoE Content Features* sublayer to the DM layer, the QoE *Characteristics* sublayer to the PM, and the *QoE Rules* sublayer to the AM. The *QoE Content Features* sublayer has the purpose of associating the metadata of every concept with a physical representation. The metadata is attached based on the concept type (e.g. a multimedia object may have associated metadata on bit rate, frame rate, resolution, number of colours, encoding, etc, whereas a picture has size, format and resolution). Metadata is used to describe the characteristics of the concept, which can be used during the adaptation process. The *QoE Characteristics* sublayer consists of classes characterising factors that have an impact on the performance. Information related to what characteristics need to be satisfied by the content in order to be delivered with success over a given network is stored at this level. The *QoE Rules* sublayer defines adaptation rules for performance-aware content selection. These rules make reference to the information contained in the QoE *Content Features* sublayer and *QoE Characteristics* sublayer. Usually, these rules are interpreted by an Adaptation Engine.

Reusable Quality of Experience aware Adaptation Strategies

In this paper we focus on the *QoE Rules* sublayer of the QoE-LAOS model. It stores rules that specify the adaptive behaviour of the e-learning system. Authoring the adaptive behaviour has been considered the most difficult component of an adaptive environment (Cristea et al. 2007, Stash et al. 2007), reusability of such components being a desirable feature (Cristea & Stewart 2006). With the aim of improving Quality of Experience, also focusing on the reusability problem, we define two adaptation strategies that aim to help to improve the learner Quality of Experience. These strategies offer the learners the best possible delivery of the educational content taking into account the learner network characteristics.

We present two strategies with the aim of giving more flexibility to the learner. One of the strategies is *adaptive* (i.e. system-driven) and the second one *adaptable* (i.e. user-driven). By using the adaptive strategy, the system automatically decides for the learner what quality content is suitable, whereas in the adaptable one, the adaptation is done based on the learner interaction with the system. The adaptive strategy may be used for less experienced learners that do not have enough knowledge about their network performance, or just want to concentrate more on the learning material, without being bothered by external factors, such as the network performance. Accurate data about network conditions are obtained in real time by an extension to the e-learning system, which continuously monitors the network parameters. Based on this information, suggestions about appropriate content quality are given. This process is managed at the *QoE Characteristics* sublayer level. The second, adaptable, strategy is for expert learners, who know the characteristics of their network and want to be in control of it. It allows the learners to choose themselves among the multimedia characteristics the one that they consider that would be the most suitable for them at that moment.

The two strategies presented are written using the LAG adaptation language (Cristea & Verschoor 2004). Different other adaptation languages have been proposed, such as LAG-XLS (Stash et al. 2007) which caters for Learning Styles. The decision of choosing LAG as the implementation language was motivated by the fact that there is no standard for adaptation languages yet, as well as by the fact that LAG represents an instantiation of the AM layer from the LAOS, authoring model, based on which QoE-LAOS is designed. A LAG code consists of two phases: an *initialization* phase and an *implementation* phase. The initialization code is executed only once. All the variables used in the implementation part need to be initialised here. Also, at this point it will be decided which concepts will be shown to the learner when the application starts. In the implementation phase, conditions and actions (e.g. PM.*GM.Concept.show* = *true*) are defined, based on which the content adaptation is done. The conditions may be a simple prerequisite (e.g. *if DM.Concept.access then PM.GM.Concept.show*) or a combination of those (e.g. conditions which contain *enough* and a number representing how many statements have to be true in order for the condition to be satisfied). The adaptation is done based on the learner interaction with the system. By default, no concept is visible in LAG, without the explicit permission of the author. Some predefined variables are defined in LAG, such as *access* and *show*. The variable *access* has a Boolean type and the value represents whether a concept is currently accessed or not. The value of the *show* variable (a Boolean type) decides whether a concept will be displayed or not. Besides the predefined variables, both strategies use variables which keep a code associated with the required value of the multimedia object parameters *(requiredBitrate*, *requiredFramerate*, *requiredResolution*, *requiredColours* and *requiredEncoding*). They represent the upper threshold of the multimedia clip parameters which may be transmitted over the network. There are as well some auxiliary variables which show whether the current concept performance attributes are suitable to be shown (e.g.*bitrateOk*, *framerateOk*, *resolutionOK*, *colursOk* and *encodingOk*). Both strategies select a multimedia from the multimedia versions available in the DM. The multimedia objects defined in the DM are different versions of the same clip but with different qualities, therefore having different performance parameters. Choosing in a controlled manner a multimedia version with a slightly lower quality to be sent to the learner, when necessary, for given network conditions, is based on the fact that this does not affect the learning process (Ghinea & Chen 2006, Muntean et al. 2008).

Authoring the content for this strategy, may be done in MOT. It allows authoring of DM and GM, based on the LAOS model. Different multimedia clips, having different performance parameters may be added through the MOT interface. In the Domain Model to every concept having a physical representation (in this case a multimedia clip), attributes have been added: bit rate (*DM.Concept.bitrate*), frame rate (*DM.Concept.framerate*), resolution (*DM.Concept.resolution*), number of colours (*DM.Concept.coulours*) and encoding (*DM.Concept.encoding*). Because MOT is a pure authoring tool, which does not allow content delivery, another system need to be chosen for this goal. The simplest solution is probably to use the AHA! system (De Bra et al. 2006, Cristea et al. 2005), an open source authoring tool developed at the Eindhoven University of Technology. The conversion between these two systems is easily done using CAF (Common Adaptation Format) files (Cristea et al. 2005). CAF file, encodes the content contained in DM and GM, and authored in MOT. The conversion engine, uses the CAF file together with the desired strategy (written in a file with the extension .lag) in order to create an AHA! compatible application.

System-Driven Adaptive Strategy

In this strategy, the recommended value of the multimedia object parameters (bit rate, frame rate, resolution, number of colours and encoding) are set by the system based on the learner network performance in the initialization phase. The auxiliary variables which need to be used during the implementation are also initialised in this phase. Conditions and actions are defined in the implementation phase. The implementation is composed of a set of conditions which are executed every time a concept is accessed. In the implementation phase is decided what concept should be shown based on the performance parameters defined during the initialisation phase.

```
initialization(
        //Required values for the multimedia parameters: bit rate, frame rate, resolution,
        //number of colours and encoding scheme. A code is associated to the resolution
        //and encoding parameter, and a mapping is done between the associated codes and
        //their values (e.g. for encoding 0 is associated with MPEG2, 1 with MPEG4 etc.).
        requiredBitrate = 1
        requiredFramerate = 25
        requiredResolution = 1
        requiredColours = 2
        requiredEncoding = 1

        //Auxiliary variables. Each variable correspond to a performance parameter (e.g.
        //bitrateOk corresponds to the bit rate).  It shows whether the current multimedia
        //characteristic satisfies or not the imposed requirements
        bitrateOk = false
        framerateOk = false
        resolutionOk = false
        coloursOk = false
        encodingOk = false
)
```

```
implementation (
        //It checks if the bit rate attribute of the concept satisfies the requirement; if
        //it does the bitrateOk variable is set to true meaning that the bit rate of the
        //concept is acceptable and the concept may be delivered over the network
        if enough (DM.Concept.access
                   DM.Concept.type == bitrate
                          , 2)
        then (if DM.Concept.bitrate <= requiredBitrate
              then bitrateOk = true
              else bitrateOk = false)

        //It checks if the frame rate attribute of the concept satisfies the requirement;
        //if it does the framerateOk variable is set to true meaning that the frame rate
        //of the concept is acceptable and the concept may be delivered over the network
        if enough(DM.Concept.access
                   DM.Concept.type == framerate
                   , 2)
        then (if  DM.Concept.framerate <= requiredFramerate
              then framerateOk = true
              else framerateOk = false)

        //It checks if the resolution attribute of the concept satisfies the requirement;
        //if it does the resolutionOk variable is set to true meaning that the resolution
        //of the conceptis acceptable and the concept may be delivered over the network
        if enough(DM.Concept.access
                  DM.Concept.type == resolution
                   , 2)
        then (if  DM.Concept.resolution <= requiredResolution
              then resolutionOk = true
              else resolutionOk = false)

        //It checks if the number of colours of the concept satisfies the requirement; if
        //it does the coloursOk variable is set to true meaning that the number of colours
        //of the concept is acceptable and the concept may be delivered over the network
        if enough(DM.Concept.access
                  DM.Concept.type== colours
                   , 2)
        then (if  DM.Concept.colours <= requiredColours
              then coloursOk = true
              else coloursOk = false)

        //It checks if the concept encoding satisfies the requirement; if it does the
        //encodingOk variable is set to true  meaning that the concept encoding is
        //acceptable and the concept may be delivered over the network
        if enough(DM.Concept.access
                  DM.Concept.type == encoding
                  , 2)
        then (if  DM.Concept.encoding <= requiredEncoding
             then encodingOk = true
             else encodingOk = false)

        //It checks if all the performance parameters satisfy all the requirements; if
        they do //then show the concept to the learner
        if enough (DM.Concept.access
                   bitrateOk == true
                   framerateOk == true
                   resolutionOk == true
                   coloursOk == true
                   encodingOk == true
                   , 6)
        then UM.GM.Concept.show = true
)
```

User Driven Adaptable Strategy

This strategy allows the learners to set the values for the performance parameters (e.g. bit rate, frame rate, resolution, number of colours and encoding) they expect, from a list of given ones. In order for the learner to be able to change the parameters to the desired ones, a menu was created and added in the e-learning system interface. Every parameter from the menu is represented in the *Domain Model* as a concept attribute. Values of the concept attribute defined in the DM are available in the menu: the user selects one attribute value. *Goal and Constraints Model* has weights and labels are associated to each concept. The labels are used to identify what parameter the learner wants to change. For example, the concept that have the value of the bit rate will have here the label *setbitrate*. The weight contains the parameter value or the code associated to the parameter value. For encoding and resolution, instead of keeping their values, a mapping is done between their codes and their values (e.g. for encoding the mapping is 1 for MPEG2, 2 for MPEG4, etc).

When the learner registers with the e-learning system, default values are given to the learner desired parameters. If the learner changes the default values, the adaptation is done taking into account the new values. The learner will get the concept with his desired parameters, if the network performance is suitable. Otherwise the leaner will get a concept which satisfies the required performance parameters.

```
initialization(
        //Required values for the multimedia parameters: bit rate, frame rate, resolution,
        //number of colours and encoding scheme. A code is associated to the resolution
        //and encoding parameter, and a mapping is done between the associated codes and
        //their values (e.g. encoding: 0 is associated with MPEG2, 1 with MPEG4 etc.).
        requiredBitrate = 1
        requiredFramerate = 2
        requiredResolution = 1
        requiredColours = 2
        requiredEncoding = 1

        //desired values for the multimedia parameters
        UM.desiredBitrate = requiredBitrate
        UM.desiredFramerate = requiredFramerate
        UM.desiredResolution = requiredResolution
        UM.desiredColours = requiredColours
        UM.desiredEncoding = requiredEncoding

        //Auxiliary variables. Each variable corresponds to a performance parameter (e.g.
        //bitrateOk corresponds to the bit rate). It shows whether the current multimedia
        //characteristics satisfies or not the imposed requirements
        bitrateOk = false
        framerateOk = false
        resolutionOk = false
        coloursOk = false
        encodingOk = false


        // Make the menu visible. All the concepts associated to the menu parameters have
        //associated a label that takes into account what parameter the learner changes;
        e.g. //the concepts that change the bit rate parameter will have the label value
        //equal to setbitrate
        while true (
                if enough (GM.Concept.label == setbitrate
                           GM.Concept.label == setframerate
                            GM.Concept.label == setresolution
                            GM.Concept.label == setcolours
                           GM.Concept.label == setencoding
                           , 1)
                PM.GM.Concept.show = true
        )
)
implementation (
        //If the learner changes the value of the bit rate with a new value from the menu,
        //change the required bit rate to the selcted one
        if enough (UM.GM.Concept.access == true
                   GM.Concept.label == setbitrate
                   , 2)
        then (UM.desiredBitrate =  GM.Concept.weight)

        //If the learner changes the value of the frame rate with a new value from the
        //menu change the required frame rate to the selected one
        if enough (UM.GM.Concept.access == true
                   GM.Concept.label == setframerate
                   , 2)
        then (UM.desiredFramerate =  GM.Concept.weight)
```

```
//If the learner changes the value of the resolution with a new value from the
//menu change the required resolution to the selected one
if enough (UM.GM.Concept.access == true
           GM.Concept.label == setresolution
           , 2)
then (UM.desiredResolution =  GM.Concept.weight)

//If the learner changes the value of the number of colours with a new value from
//the menu change the required number of colours to the selected one
if enough (UM.GM.Concept.access == true
           GM.Concept.label == setcolours
           , 2)
then (UM.desiredColours =  GM.Concept.weight)

//If the learner changes the value of the encoding with a new value from the menu
//change the required encoding to the selected one
if enough (UM.GM.Concept.access == true
           GM.Concept.label == setencoding
           , 2)
then (UM.desiredEncoding =  GM.Concept.weight)


//It checks if the bit rate attribute of the concept satisfies the requirement; if
//it does the bitrateOk variable is set to true meaning that the bit rate of the
//concept is acceptable and the concept may be delivered over the network
if enough (DM.Concept.access
           DM.Concept.type == bitrate
          , 2)
then (if UM.desiredBitrate <= requiredBitrate
      then (if DM.Concept.bitrate == UM.desiredBitrate
            then bitrateOk = true
            else bitrateOk = false)
      else (if DM.Concept.bitrate <= requiredBitrate
           then bitrateOk = true
           else bitrateOk = false))

//It checks if the frame rate attribute of the concept satisfies the requirement;
//if it does the framerateOk variable is set to true meaning that the frame rate
//of the concept is acceptable and the concept may be delivered over the network
if enough(DM.Concept.access
          DM.Concept.type == framerate
          , 2)
then (if  UM.desiredFramerate <= requiredFramerate
      then (if DM.Concept.framerate == UM.desiredFramerate
            then framerateOk = true
            else framerateOk = false)
      else (if DM.Concept.framerate <= requiredFramerate)
            then framerateOk = true
            else framerateOk = false))

//It checks if the resolution attribute of the concept satisfies the requirement;
//if it does the resolutionOk variable is set to true meaning that the resolution
//of the conceptis acceptable and the concept may be delivered over the network
if enough(DM.Concept.access
          DM.Concept.type == resolution
          , 2)
then (if  UM.desiredResolution <= requiredResolution
      then (if DM.Concept.resolution == UM.desiredFramerate
            then resolutionOk = true
            else resolutionOk = false)
      else (if DM.Concept.resolution <= requiredResolution
           then resolutionOk = true
           else resolutionOk = false))

//It checks if the number of colours of the concept satisfies the requirement; if
//it does the coloursOk variable is set to true meaning that the number of colours
//of the concept is acceptable and the concept may be delivered over the network
if enough(DM.Concept.access
          DM.Concept.type== colours
          , 2)
```

```
        then (if  (UM.desiredColours <= requiredColours
              then (if DM.Concept.colours == UM.desiredColours
                    then coloursOk = true
                    else coloursOk = false)
              else (if DM.Concept.colours <= requiredColours
                    then coloursOk = true
                   else coloursOk = false))

        //It checks if the concept encoding satisfies the requirement; if it does the
        //encodingOk variable is set to true  meaning that the concept encoding is
        //acceptable and the concept may be delivered over the network
        if enough(DM.Concept.access
                  DM.Concept.type == encoding
                   , 2)
        then (if  (UM.desiredEncoding <= requiredEncoding
              then (if DM.Concept.encoding == UM.desiredEncoding
                    then encodingOk = true
                    else encodingOk = false)
             else (if DM.Concept.encoding <= requiredEncoding
                   then encodingOk = true
                   else encodingOk = false))


        //It checks if all the performance parameters satisfy the requirements; if they do
        //then show the concept to the learner
        if enough (DM.Concept.access
                   bitrateOk == true
                   framerateOk == true
                   resolutionOk == true
                   coloursOk == true
                   encodingOk == true
                   , 6)
        then UM.GM.Concept.show = true
)
```

Discussions

The presented adaptation strategies for multimedia clips, may be re-used, with small modifications, in order to perform adaptation based on other multimedia parameters, or even a combination of them. The strategies, as presented in this paper, may be simply re-used by every author who wants to improve the learner QoE (as long as the LAG language is supported by the authoring tool or there is a way to make a conversion between authoring tools).

The advantage of defining these strategies in the LAG language and using them with MOT is that MOT is compatible with other authoring systems (Power et al. 2005). The strategies may be used as they are and re-used between different domain models without the need to change them. However, up to now, the performance attributes are introduced manually in the Domain Model. This increases the work the author needs to do when creating performance aware courses. Currently we are working on overcoming this problem, and on automatically process for adding the performance attributes when a multimedia is introduced in DM will be developed

The strategies should be used, to assure that every learner has access to the educational content, even when the network performances are poor. Differences in network performance “exists between the developed and developing countries, but also between urban and rural regions within every country” (Johnson et al. 2007). This may lead to learners which have access to learning material and the ones that wait for a long time until the educational content is displayed on the screen or they receive educational content with the integrity affected.

Conclusions and Future Work

This paper presented two Quality of Experience aware adaptation strategies to be used with authoring tools. The strategies take into account the concepts’ performance attributes as well as suggestions regarding desirable characteristics for the content to be transmitted over the learner's network. Using these strategies, during the adaptation process, may help learners' to access educational content regardless of their network performance. These strategies may be re-used as they are suitable for any course content. They ease the author work and provide adaptation support to deliver courses for any students, indifferently how network connectivity do they have.

We are currently implementing an automatic mechanism to detect the multimedia performance characteristics and to introduce this metadata in Domain Model of the MOT application. We are concerned as well with mechanisms to detect the network conditions in real time and to provide suggestions on the most suitable content characteristics. Subjective tests need to be performed, to evaluate the use of the authoring tool and to analyse learners' perception of a course created based on these strategies.

References

Aroyo, L., Cristea, A.I., & Dicheva, D. (2002). *A Layered Approach Towards Domain Authoring Support*, Proceedings of the International Conference on Artificial Intelligence, IC-AI 2002, Las Vegas, USA.

Brusilovsky, P. (2003). *Developing Adaptive Educational Hypermedia Systems: From design models to authoring tools*. In: T. Murray, et. al (eds.): Authoring Tools for Advanced Technology Learning Environment. Dordrecht: Kluwer Acad. Publishers, 377-409.

Cristea, A., & Calvi, L. (2003). *The Three Layers of Adaptation Granularity*. In User Modeling 2003. Retrieved August 10, 2009, from http://dx.doi.org/10.1007/3-540-44963-9_4.

Cristea, A. I., & de Mooij, A. (2003). *LAOS: Layered WWW AHS Authoring Model and their corresponding Algebraic Operators,* 2003, The Twelfth International World Wide Web Conference, Budapest, Hungary.

Cristea, A. I., Smits, D., & De Bra, P. (2005). *Writing MOT, Reading AHA! - converting between an authoring and a delivery system for adaptive educational hypermedia* - A3EH Workshop, AIED'05 July, 2005, Amsterdam, the Netherlands

Cristea, A. I., Smits, D., & De Bra, P. (2007). Towards a Generic Adaptive Hypermedia Platform: a Conversion Case Study, *Journal of Digital Information, (JoDI), Special Issues on Personalisation of Computing & Services*, 8(3), Retrieved August, 10, 2009, from http://journals.tdl.org/jodi/article/view/231/184.

Cristea, A.I., & Stewart, C.D. (2006). *Automatic Authoring of Adaptive Educational Hypermedia*, book chapter II in Web-based Intelligent E-Learning Systems: Technologies and Applications", ZongMin Ma (Ed.), Info. Science Publishing (IDEA group), 24-55.

Cristea, A.I., & Verschoor, M. (2004). *The LAG Grammar for Authoring the Adaptive Web*, Proceedings of the International Conference on Information Technology: Coding and Computing, ITCC'04, April, 2004, Las Vegas, US.

De Bra, P., Smits, D., & Stash, N. (2006). Creating *and Delivering Adaptive Courses with AHA!*, 2006, Proceedings of the first European Conference on Technology Enhanced Learning, LNCS, Crete, 21-33.

Ghinea, G., & Chen, S. Y. (2006). Perceived Quality of Multimedia Educational Content: A Cognitive Style Approach, *ACM Multimedia Systems Journal*, 11(3), 271-279.

Hooper, S., Rose, S., Miller, C., and Veletsianos, G. (2007). The Effects of Digital Video Quality on Learner Comprehension in an American Sign Language Assessment Environment. Sign Language Studies, 8(1), 42-58.

Johnson, R., Kemp, E., Kemp, R., & Blakey, P. (2007). The Learning Computer: Low Bandwidth Tool that Bridges Digital Divide. *Educational Technology & Society*, 10(4), 143-155.

Karampiperis, P., & Sampson, D. (2005). Adaptive Learning Resources Sequencing in Educational Hypermedia Systems. *Educational Technology & Society*, 8 (4), 128-147.

Kuo, Y.-H., & Huang, Y.-M. (2009). MEAT: An Authoring Tool for Generating Adaptable Learning Resources. *Educational & Technology & Society*, *12* (2), 51–68.

Kumar, K. (2005). *A Marriage made in QoE Heaven*. CED Magazine (July 2005), 37-39.

Lukasenko, R., & Grundspenkis, J. (2009). *A Conception of Agent-based User Modelling Shell for Intelligent Knowledge Assessment System*, 2009, IADIS e-Learning, Algarve, Portugal, 98-106.

Manzoor, S., Libbrecht, P., Ullrich, C., & Melis, E. (2005). Authoring Presentation for OpenMath, *Proceedings of the 4th International Conference on Mathematical Knowledge Management (MKM 2005),* Bremen, Germany.

Molnar, A., & Muntean, C.H. (2009). *QoE MOT – A Learner Quality of Experience-Oriented Authoring Tool*, European Conference on Technology Enhanced Learning, A3H: Seventh International Workshop on Authoring of Adaptive and Adaptable Hypermedia, 29 Sept.-2 Oct. 2009, Nice, France.

Muntean, C. H. (2008). Improving Learner Quality of Experience by Content Adaptation based on Network Conditions, *Computers in Human Behavior*, *Special Issues on Integration of Human Factors in Network Computing*, 24(4), 1452-1472.

Muntean, G.-M., Ghinea, G., & Sheehan, T. N. (2008). Region of Interest-based Adaptive Multimedia Streaming Scheme*, IEEE Transactions on Broadcasting*, 54(2), 296-303. doi: 10.1109/TBC.2008.919012

Muntean, C.H., Muntean, G.-M., McManis, J., & Cristea A.I. (2007). Quality of Experience - LAOS: Create Once, Use Many, Use Anywhere, *International Journal of Learning Technology (IJLT), Special issue on Authoring Adaptive and Adaptable Hypermedia*, 3(3), 209-229.

Odence, P. (2004). *Assuring QoE on next generation networks*, Annual Review of Communications, International Engineering Consortium, White paper, Empririx, November, Vol.57, http://www.katech.fi/tiedostot/empirix/NGNT_AssuringQoE.pdf.

Power, G., Davis, H. C., Cristea, A. I., Stewart, C., & Ashman, H. (2005) Goal Oriented Personalisation with SCORM. In: *The 5th IEEE International Conference on Advanced Learning Technologies (ICALT 2005)*, July 5 - 8, 2005, Kaohsiung, Taiwan.

Stash, N., Cristea, A. I., & De Bra, P. (2007). Adaptation Languages as Vehicles of Explicit Intelligence in Adaptive Hypermedia. *International Journal of Continuing Engineering Education and Life-Long Learning*, 1 (4/5), 319 - 336. doi: 10.1504/IJCEELL.2007.015045.

Sleeswijk Visser, F., van der Lugt, R., & Stappers, P.J. (2007). Sharing User Experiences in the Product Innovation Process: Participatory Design needs Participatory Communication. *Journal of Creativity and Innovation Management*, 16(1), 35-45.

Te-Hua Wang, Yen, N., Yue-Lin Du, & Shih, T. (2007). Courseware Authoring Tool for Achieving Interoperability among Various E-Learning Specifications Based on Web 2.0 Technologies. In *Parallel Processing Workshops, 2007. ICPPW 2007. International Conference on* (p. 25). Presented at the Parallel Processing Workshops, 2007. ICPPW 2007. International Conference on. doi: 10.1109/ICPPW.2007.32.

Yudelson, M., Medvedeva, O. & Crowley, R. (2008). A Multifactor Approach to Student Model Evaluation, *User Modelling and User-Adapted Interaction*, 18(4), 349-382.

Acknowledgements

This research work was supported by IRCSET Embark Postgraduate Scholarship Scheme and SFI Research Frontiers Programme.