HEURISTICS SUPPORTING USABLE AUTHORING TOOLS
Matching the right tool to the right user

Paula Kotzé, Elsabe Cloete
School of Computing, University of South Africa, Pretoria, South Africa

Keywords: E-learning, User needs, Functional and non-functional requirements, Interfaces and usability, Reusability, Instructional authoring tools

Abstract: Over the past few years while e-learning has been gaining momentum, the user profile of instructional authoring tools has also evolved. It seems that commercial authoring products have not yet been adapted to address all user groups, which impedes lecturers in their working environment while preparing e-learning materials, with the materials not achieving the required quality as a result. In this paper heuristics to design an authoring tool aimed at a specific user group, namely the ordinary lecturer, are described to enable subject-expert lecturers (not necessarily technically skilled) to create and reuse their own e-materials without undergoing intensive technical training. The significance of these heuristics lies in the fact that they provide a method to overcome many of the complexities associated with the design of instructional authoring tools. Furthermore, tools developed according to these heuristics might enable institutions to cope with the universal design demands associated with e-learning, without their e-learning programmes being delayed by the scarcity of professional instructional designers and instructional programmers.

1 INTRODUCTION

Authoring support environments (ASEs) (also called ‘instructional authoring tools’) are mission-critical applications in achieving usable e-learning materials. Developing e-learning materials requires competencies in several areas, including subject matter, pedagogical foundations, instructional systems development, and also ASE experience (Tennyson, 2001). Cramer (2003) defines three business application categories of ASEs that are aimed at different user groups:

– ASEs complying with recognized instructional design principles, requiring all the abovementioned competencies;
– ASEs that do not adhere to instructional design practices, but create materials that may or may not have a training focus; and
– entry-level ASEs offering restricted functionality and limited deployment options without much instructional design at the front end.

Ideally, educational institutions, and specifically universities, should employ enough professional instructional designers who can, together with instructors, develop e-learning materials. In reality the paucity of these professionals combined with the potentially large number of courses (sometimes thousands), often necessitate instructors having to take responsibility for the development of part, or all of their e-materials. Furthermore, some intermediate communication/content/errata may be so diminutive or urgent that it is not always possible to involve an instructional design team in its composition, but the lecturer is nevertheless obliged to handle its introduction into the e-classroom. Since instructors are often subject matter experts as well as being trained in pedagogical approaches, we find that there is a fourth ASE user group, namely those subject experts who have educational foundations and intuitive strategies to promote learning, but possibly lack technical experience. This group requires an efficient and usable ASE, simple in its approach but without a corresponding loss of instructional quality (Miller, 2003).

The question that we address in this paper is how the usability of ASEs and the reusability of their output could be improved in order to enable subject-expert instructors, who are not necessarily technically skilled, to create and reuse their own e-materials without undergoing intensive technical training.

Literature provides many guidelines on the usability aspects of e-learning materials (see for example IDE (2002), Macromedia (2002), Mehlen-

Kotzé P. and Cloete E. (2004).
HEURISTICS SUPPORTING USABLE AUTHORING TOOLS - Matching the right tool to the right user.
In *Proceedings of the Sixth International Conference on Enterprise Information Systems*, pages 169-178
DOI: 10.5220/0002612901690178

bacher (2001), National Cancer Institute (n.d.), and Step Two Designs (2002). However, design guidelines to improve author usability in ASEs are limited. The W3C (2000) recommends guidelines for web authoring software developers to assist them in designing ASEs that could produce accessible web content, and also be accessible to their users. These guidelines describe *which* usability issues are to be addressed, while our discussion focuses on *how* usability parameters can be introduced in ASEs. Furthermore, the target user group of web authoring tools assumes technical experience, which we cannot assume for our target user group.

According to Duchastel (2001) information design is an abstraction process determining *what to include and what to leave out;* in effect determining at what level and how to present specific content, in this case, to the ASE author. The problem at hand is complex due to:

– the number of ASE elements to consider;
– the fact that many of these elements are amalgamated with the courseware elements, which are aimed at a different target user group; and
– the fact that many of these elements are complex, being embedded in other elements.

What makes it complex to integrate usability principles into ASEs, is the separation and isolation of the amalgamated elements. One has only to consider the various worldwide efforts to create usable ASEs (based on open educational standards) to become aware of the complexity of the situation (De Vries, 2002). In this paper, we describe how to disentangle the amalgamated elements that are integral to both the authoring software and courseware products. We believe by isolating the elements it is possible to go beyond superficial design issues and apply suitable usability parameters to the isolated elements, before integrating them into the authoring environment.

The significance of the heuristics that we have established and report on in this paper, lies in their provision of a method of overcoming many of the complexities associated with the design of ASEs for this specific target user group. Furthermore, a tool that is developed according to these heuristics will enable institutions to cope with the design demands associated with e-learning, without their e-learning programmes being stalled by the scarcity of professional instructional designers. We do not claim that the heuristics or the results are final or complete. We have implemented them in a prototype with limited functionality and share some of the preliminary results. We recognize that further testing needs to be done.

The entire e-learning domain forms an integrated unit, and research focus on one level may easily fan out to other levels, and direct or specify how these levels should behave. Currently, there are several standardisation efforts attempting to standardise different aspects of the e-learning environment, as this will improve coordination, reuse, interoperability and vendor specifications considerably. As the focus of our research is the creation of courseware through a suitable tool, we briefly refer to accomplishments that are focused on the building blocks of courseware and their relation to our work, before we commence the detailed discussion of our work. The most prominent international standardisation efforts for courseware elements include IMS/EML (EML 2001, IMS 2002), IEEE/LOM with Dublin Core (IEEE, n.d.), SCORM (ADL, n.d.) and ARIADNE (n.d.). The purpose of these standards is to describe information. For the focus of our research, the purpose is to describe learning objects so that they are independent, reusable and easily retrievable units. The purpose of our research is therefore focused on how to design a lesson into which such units are integrated. There is therefore a similarity in certain terminology, since the basic building blocks are the same, yet there is dissimilarity where our research focus swings away from the building blocks to the method of creating them.

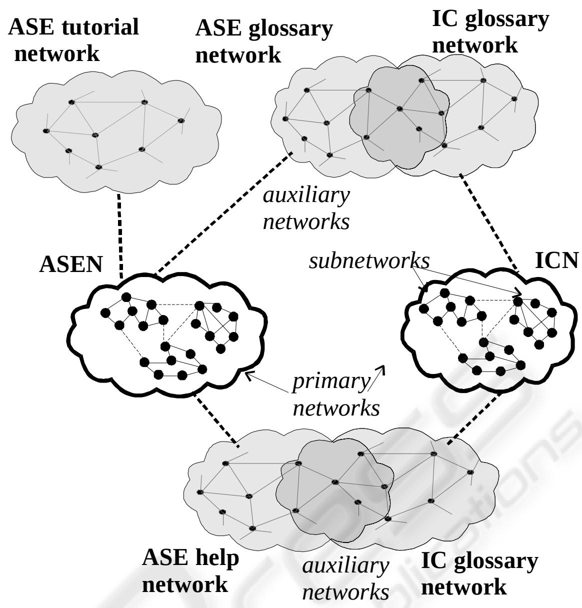

Figure 1: Internetwork of associated e-learning networks

Section 2 of this paper introduces a characterisation of e-learning networks forming the conceptual basis for the discussion that follows. Section 3 is an exposition of the aforementioned design requirements. These requirements are used in Section 4 to depict a design framework for ASEs adhering to reusability and usability principles. In the same section we discuss how we used this framework to implement our ASE prototype. Evaluation of the prototype is commented on in Section 5, and conclusions are drawn in Section 6.

2 E-LEARNING NETWORKS

The e-learning environment can be considered as an internetwork structure where different networks are comprised of network segments, which in turn are composed of associated nodes connected via links.

Nodes are functionally defined units representing content. Categorising these nodes reduces the complexity of the network. Typical category types include glossary-type, help-type, question-type, annotation-type, simulation-type, discussion-type, browsing-type, and termination-type nodes (Kotzé, 1997). The names of these categories are self-explanatory and as they suggest, these categories provide a method of formalising content.

Links depict the potential routes between the nodes within a network. As with nodes there are also different link category types, including contextual, referential, detour, annotational, return and terminal links (Kotzé, 1997). *Contextual links* provide a route-tracking mechanism recording the user's path through the network. *Referential links* are typically the hyper-links that appear on an interface, allowing the user to pursue different routes through the network. *Detour links* connect the user to another (usually auxiliary) network. *Annotational links* allow the user to connect to an annotation node to record personal notes. A *return* link returns from a diversion initiated by a detour or annotational link, and finally *terminal links* indicate that the end of the current network has been reached.

Two primary networks are portrayed in Figure 1, namely the *Authoring Support Environment Network* (ASEN) and the *Interactive Courseware Network* (ICN). The principal users of ASENs aim to create courseware that produces an ICN for deployment on CD, the Intranet or the Internet to be used by the principle users of the ICN, namely the learners. Figure 2 illustrates the input-output parameters of the main networks, as well as their relation to one another. The primary user interacts with the ASEN by defining the different learning objects by using plain text, graphics, defining links, et cetera. The output of the ASEN then becomes the ICN, which is used by both the primary ICN user (the learner) and the secondary ICN user (the instructor).

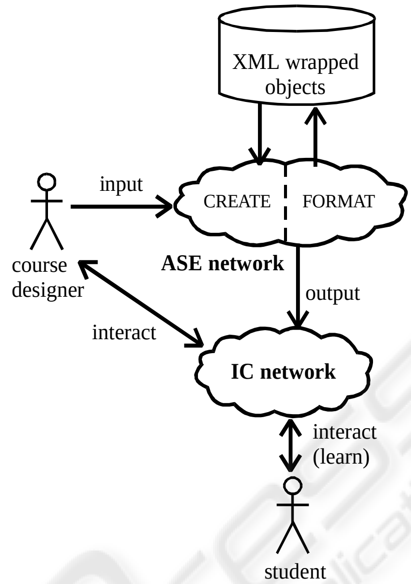

Figure 2: Input/output parameters of ASEN and ICN

The internetwork (illustrated in Figure 1) consists of a set of states as well as relationships between the states and allowable operations on these states. We associate two states with each network; namely an *internal state* and an *external display state*. The internal state portrays the position or condition of the nodes under development, as well as the variables that are required to control interaction with the author while creating and interacting with these nodes. For example, while a course designer creates a question-type node, values such as the type of node being created, special attributes for the particular node and conditions before the question or parts thereof (like its answer) may be displayed, are set. The external display state, on the other hand, is the perceived network interface with which the actor interacts. This state reflects parts of, but not necessarily the entire internal state. For example, the student who interacts with the question-type event observes the content and certain conditions that pertain to the question, but not the answer. The completed ICN result state is exported for delivery on a specific platform.

Figure 3, adapted from Kotzé (1997), illustrates the mappings between the internal and external display states within each network, as well as the relationships among them. The interaction of the actor with the ASEN nodes is reflected in mapping (a) between the ASEN's internal and external states. The internal state of the ASEN also maps to a resulting ICN state in mapping (b) that shows how the resulting ICN is reflected during the authoring process. Furthermore, the resulting state of the ICN

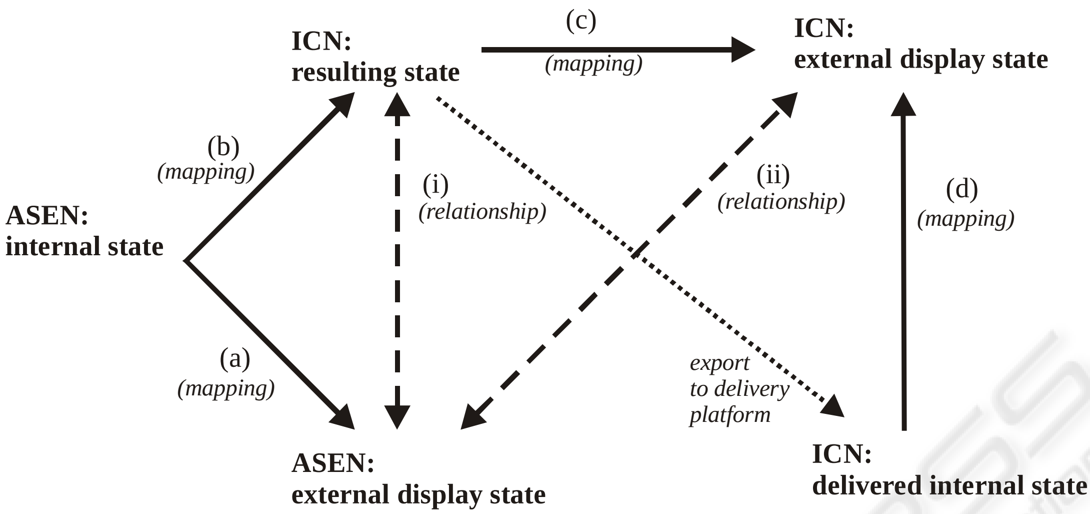

Figure 3: Mapping and relationships between ASEN and ICN

has an externally perceivable rendering of the ICN during the authoring process, as illustrated in (c). Once the ICN has been exported to the required delivery platform (during the learning process), the internal state of the ICN maps to its external display state illustrated in (d) (in the same way as for the ASEN during the authoring process). The dotted lines (i) and (ii) indicate that there is a relationship between the different states in the ASEN and ICN. As mentioned before, in this paper our focus and interest lie in the ASE network. However, Figure 3 illustrates that it is impossible to consider the ASEN in isolation since the states in this network have a direct impact on the resulting ICN system.

There are also a number of auxiliary networks attached to each of the primary networks to assist the users in their interaction with the primary network. For example, the ASEN in Figure 1 has three auxiliary networks attached to it, namely the *glossary network*, the *help network* and the *tutorial network*. The user's interaction with these networks may enhance focus and efficiency, but is not an essential element in achieving the user's primary goal. In fact, expert users seldom interact with these types of networks. In many cases there are overlaps between different auxiliary networks. These overlaps are often non-deterministic, which means that entrance from one network into another might imply that a user could be trapped in a situation where the network never terminates.

3 DESIGN REQUIREMENTS

Several sources were consulted and used to establish tangible usability criteria for an ASEN interface (ASTD, 2002; Badre, 2002; Cloete, 2003; Dix, 2004; Duchastel, 2001; Kotzé, 1997; Preece, 2002; Proctor, 2002). These criteria are summarised below and are helpful guiding principles to introduce acceptable levels of usability in the ASEN interface (ASENI). The criteria are organised according to the basic principles of usability, namely learnability, flexibility and robustness, and are then further extended to the individual elements associated with each.

Learnability is defined as the measure of how easy it is to productively begin to use an interface producing the desired results (Proctor, 2002). Learnability elements of interest to the ASENI include:

– Predictability: How consistent is the ASENI with the author's expectations?
– Familiarity: How well can the author relate the ASENI interaction to his usual methods of preparing learning materials?
– Generalisability: Is the degree of consistency between the different ASENI elements high enough to enhance the author's predictability of the interface?
– Internal consistency: Are there elements in the ASENI that distract the author?
– Subjective satisfaction: Does the author like using the interface?

Flexibility refers to the number of ways in which the user can interact with the ASENI (Dix, 2004).

Usability parameters of interest that contribute to the flexibility of the ASENI include:

– Dialogue initiative: Does pre-emptiveness lie with the author or the ASEN?
– Multi-threading: Does the ASENI provide an intuitive environment where the author can work in multiple windows or on multiple tasks?
– Navigational functions: Can the author move through the interface at own choosing?
– Task migratability: Can the author trust the ASE to automate certain functions while taking responsibility for others of his or her own choosing?
– Customisability: Does the ASENI include formatting, presentation, legibility options, as well as WYSYWIG view capabilities?
– Orientation and tracking: Does the ASENI include synthesisability features to track the movements of the author, enabling him to orientate himself with regard to the ASEN?

The robustness of a network refers to the features supporting the successful achievement and assessment of the goals (Dix, 2004). Usability parameters affecting the robustness of an ASENI include:

– Observability: To what degree can the author readily determine the working of the ASE and the interface?
– Error frequency, severity and recoverability: What is the frequency with which the author makes errors? How serious are these, and how can they be recovered?
– Responsiveness: Does the ASENI give timely feedback?
– Task conformance: To what degree does the ASENI tasks comply with the intended actions of the author?

4 DESIGN FRAMEWORK

The design decisions made during interaction with the ASENI are pertinent to the ICN interface (ICNI). Owing to this relationship between them, as illustrated in Figures 1 to 3, it is not possible to isolate the design criteria for the ASEN from the design criteria of an ICN. Our proposed design guidelines consist of four phases as set out in the subsections to follow. However, before describing each of these phases and how we implemented them, we comment on the ASE prototype that we developed according to these phases.

We developed an ASE prototype of limited scope, meaning that the prototype covered only one learning unit (see Section 4.1) as a subnetwork of the primary ASEN. The objectives of this prototype were twofold, namely to provide an authoring tool to develop an ICN that is universally usable given the target audience, as well as being able to create learning objects that are based on open standards to foster large-scale reuse. We have limited the scope of our prototype to prove that the suggested objectives are indeed achievable within such an ASEN.

The prototype was developed so that on first-time use, the author is prompted with the opportunity to enter personal and general course details such as course facilitator, course code and so forth. Although this does not initially promote user pre-emptiveness, it sets the scene for enhanced orientation and tracking capabilities, adaptability, and task migration.

4.1 Phase 1: Identify Learning Units

The initial focus of the design is on the content of the learning situation, thus on the ICNI. The purpose of Phase 1 is to provide the author with a tool to identify manageable and sensible chunks of content, which would typically be handled in one learning session. These chunks are called *learning units* (LUs). An example of LUs can be found when considering a course, say *Systems Analysis*. Typical LUs of this course might include basic concepts, requirements gathering, requirements validation, dynamic modelling, class modelling, and so on. Defining LUs is therefore a fairly generic process, not subject-specific, but merely a mechanism to provide the author with the means to identify different subject-specific chunks of content.

The term ‘learning unit’ correlates with the term used in learning environments in general, which increases familiarity when used in the ASENI. Once the author has grasped the intended meaning of an LU (it should actually be part of his/her pedagogical foundation), constructing the courseware with LUs bears a resemblance to designing a paper-based lesson. Therefore, the use of LU elements in an ASENI increases both the predictability and generalisability of the interface because the author is faced with a customary lesson design environment.

Implementation of LUs typically means the inclusion of a menu option such as <*Define new LU*>. This type of menu item (or interface button) shifts the perceived pre-emptiveness from the interface to the author. The basic interface layout of our ASEN prototype, however, only contains buttons to define, maintain or delete topics (see Section 4.2) and events (see Section 4.3), since we limited the scope of the prototype to include only one LU, namely L_1=COURSE INFORMATION.

Even though we created only one LU, we found that users had an inherent understanding of the concept of an LU. The specific articulation of distinct LU elements (for example in a separate window) and how they are linked together, simplifies the inclusion of multi-threading and navigational functions, whilst observability is increased, with the author being able to perceive a clear view of the course elements and hence the course structure.

4.2 Phase 2: Topic Identification

Once again in this phase, we identify, isolate and formalise ICN elements, but this time we focus on the main topics within a specific LU. The focus in this phase therefore remains on the content. Returning to our previous example of the Systems Analysis course and using the *Basic Object-Oriented Concepts* LU, we can, for example, identify the main topics in this phase as being objects and classes, class attributes, object and class relationships, methods, encapsulation, polymorphism, inheritance, generalisation and specialisation, and so forth. Each topic forms a network segment of the ICN. Formally, each LU is comprised of a set of topics that are epitomized by that particular LU:

$$\forall LU_k \exists \{T_{kj}\}_{j=1}^{m}, \text{ where k, } j, m \text{ are natural numbers}$$

As with the LUs, the articulation of topics in this way increases predictability, familiarity, generalisability, multi-threading, and the inclusion of navigational functions. Furthermore, if the system can be trusted to order the topics within the different LUs, with the option that the user may change this order by drag and drop activities, task migratability can be promoted.

In our ASE prototype, we predefined a number of specific topics. In general, topic selection should not be content-specific, but since the topics of the selected LU_1 ('Course Information') are universal for all courses, we specified the following topics:

T_{11} = OBJECTIVES	T_{16}=SYLLABUS
T_{12}= PREREQUISITES	T_{17}=HOW_TO_STUDY
T_{13} = MATERIAL	T_{18}=INTERNET_ACCESS
T_{14}= COMMUNICATION	T_{19}=ASSIGNMENTS
T_{15} = ASSESSMENT	

4.3 Phase 3: Event Declaration

4.3.1 Main networks

During the third phase, the design focus of the ICN merges with the design focus of the ASEN. Teaching and learning occurs during the interaction of various learning events such as questions, self-tests, discussion sessions, and so forth. Our focus therefore shifts to the methods conveying the content. Our intention is to create a set of all possible methods, and make them available to the author, giving the author freedom to select one or more events appropriate to render possible the learning of a specific topic. This can be done by first considering a set of events that are associated with each topic, and then uniting all event sets so as to capture the set that underpins the possible learning methods. Each event becomes a node in the network segment of the specific topic. We have defined such a set of events as including:

E = {discussions, questions, examples, exercises, self-tests, simulations, URIs} (URIs = Uniform Research Identifiers, for example, a link to an external resource such as a diagram, video clip, URL (Uniform Research Locator – referring to a web address), et cetera.)

ASENI designers can add to this set, but it is advisable to guard against a too fine-grained categorisation of events, as this might introduce interaction complexities, especially where unfamiliar terminology is used that focuses on computing technologies and terminology rather than on learning terminologies.

Each event has specific attributes making the event flexible for use in different circumstances. In the ICN, attributes play an important role in constituting the learning environment, while in the ASEN, the *presentation* of each attribute and how the ASE user can interact with it, contributes to the usability of the ASEN. To illustrate the point, we briefly describe one type of event, the question-type event, in more detail. For a detailed explanation of all event descriptions, see Cloete and Kotzé (2003).

A *question-type* event is defined with customisable attributes and can be constructed by setting several attributes, of which only a few are compulsory while the others default to NIL. We define a *question-type* event as follows:

$$E_{quest} = E_{compound} \cup Q_j \textit{ where}$$
$$Q_j = \{no, QuestionPos, show, cmnt, qst, ans, hint, crd\} \cup Qi$$
$$\textit{and } Q_i = \{opt, optValue\}$$

with $E_{compound}$ *a compound event composed of other events, that together form an independent event which can act as a network node.*

The first two attributes refer to the question number and whether it is a main or sub-question – allowing for task migratability, where the user can either expect the system to handle the numbering or use his/her own numbering scheme. The *show* attribute is used to either display or withhold the answer, depending on other criteria that the author has. The author can add comments that are not perceivable in the ICN by using the *cmnt* attribute. The *qst*, *ans*, *hint*, and *crd* attributes refer to the question itself, the answer, any hints that are to be displayed in the ICNI, as well as the credits for the question. For multiple-choice questions, the user can activate the *opt* and *optValue* attributes. To enhance customisability and learnability during implementation, the network should initially respond to the user's request to create a question node by producing a perceivable coherent interface window where only compulsory attributes await input. A *<More Advanced..>* button can then give the user access to the other attributes. Keeping a counter to determine how often the user interacts with the *<More Advanced..>* button, the ASENI can adapt the perceivable window to make the most frequently accessed (or all) attributes available on the same interface window as the compulsory attributes.

Owing to the limited scope of our ASE prototype, we restricted the available events on the interface to include *discussion-type*, *exercise-type*, *question-type*, and *link-type* events. We briefly explain how we associated these events with the topics discussed in the previous phase. When the author defines (selects) a new topic, one or more events are either associated with it, or the author is given the option to select specific events to associate with the new topic. Whether this association is hard-coded or created by the author, depends on the type of topic. For example, the events associated with T_{11} through to T_{18} are hard coded as a combination of discussion-type and link-type events since these topics contain (flat) content that has to be presented to the user, with little or no interaction expected from the user. However, for T_9 ('*assignments*'), the author has the option of associating different, or a combination of event types with each assignment that is defined. As such, the author might start an assignment (*exercise-type event*) by creating a scenario (*discussion-type event*), followed by references (*link-type events*), before stating the problem (*question-type event*). Figure 4 represents a screenshot from the ASEN prototype showing how the exercise-type event is depicted.

4.3.2 Auxiliary networks

The structure of an auxiliary network is functionally integrated with its primary network. Although it is an autonomous network, it cannot be designed to be entirely independent of, or in isolation from, its primary network. We briefly mention three archetypal nodes for the different auxiliary networks and mention the related usability aspects. The *troubleshooting* event assists one to find the reason for inexplicable behaviour of the network and also to find measures to improve it. A help event provides an explanation of the purpose of another event or terminology. A *tutorial* consists of several links referring mainly to different *discussion*, *simulation* and *help* events. The auxiliary networks thus add to the robustness of the ASENI, specifically enhancing observability and recoverability. Task conformance is enhanced as help and simulation events can explain the purpose of the task should the author misunderstand it. We have not integrated any auxiliary networks into our prototype at this stage.

4.4 Phase 4: Link plotting

After definition of the network nodes, network routes are designed and specified in a navigational table. In a simple situation, the navigational table can be a simple indexed database. However, in a fully developed network environment where the primary networks are augmented with auxiliary networks, is where a mesh of logical routes possibly exists. In this case an improved store-and-retrieval method is required, where entries in the navigational table are stored in pairs of the format (*s*,*n*), where *s* refers to the source node's ID and *n* refers to the next node's ID. A next-hop routing algorithm uses a one-step-along-the path approach to identify the next node en route to the destination. As a first step in determining a route from source to destination, possible routes from the source are extracted. If none of these provide a direct link to the destination, entries that include the destination address are extracted next, and their sources are followed backwards until the shortest route from the source is determined.

A design challenge is to provide a suitable tool for route visualisation, route planning and route creation. This can be achieved by creating an environment where created nodes are displayed and the author can visually connect nodes. Observability, orientation and tracking are greatly enhanced by making the completed mesh graph perceivable to the user. However, for large routing tables, the complete mesh may actually increase complexity of the interface instead of simplifying it. In such a case, flexibility parameters such as adaptability and adaptivity should be given special attention during the ASENI design, by making provision for the interface to expose only a route cluster at a time, instead of the complete mesh.

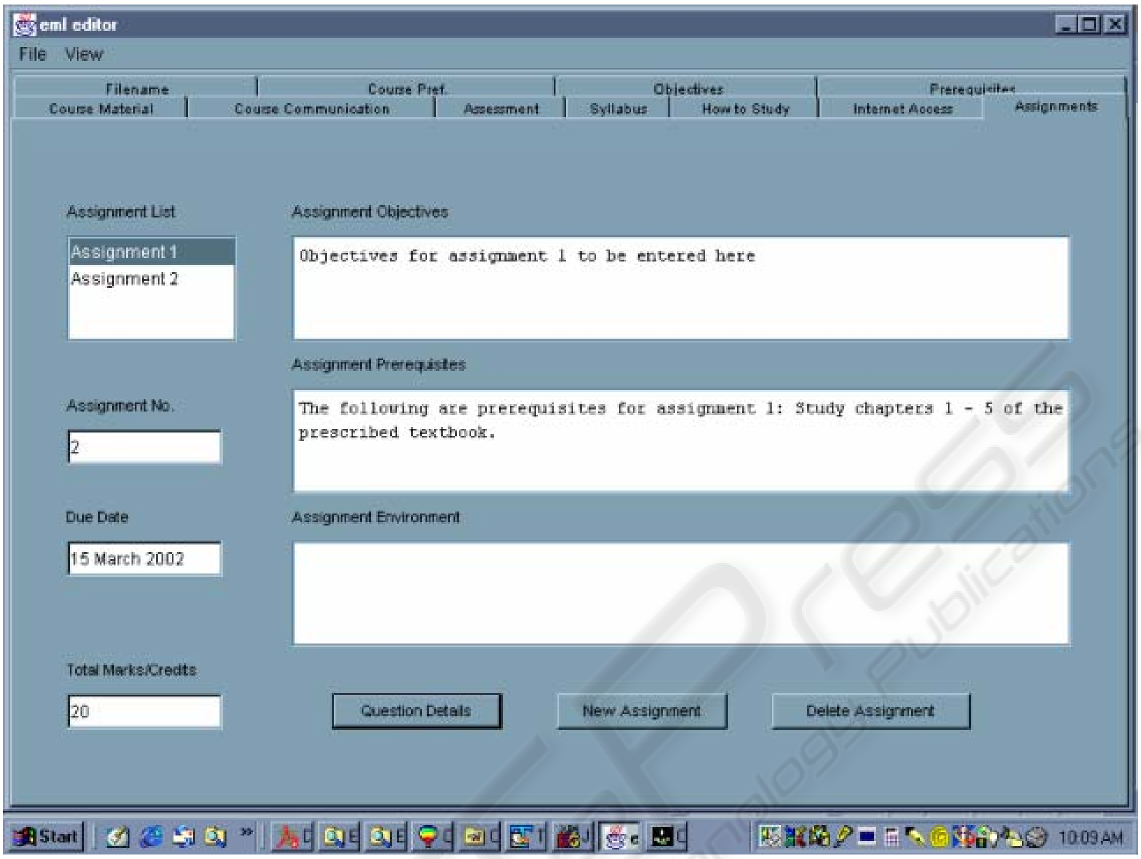

Figure 4: ASENI prototype screenshot of *Exercise* environment

Navigation through the ASEN prototype is straightforward, following the indexed database approach.

4.5 Phase 5: Exporting and Delivery

The complicated part of developing the prototype was to export the author's content to an appropriate standard output format after the authoring process. The aim is also to export the learning objects to an open standard that would enable reuse, without burdening the user with the technical details of the required standard.

We refrained from exporting the content during the authoring process, as editing and multi-threading capabilities would have complicated the programming task tremendously.

The export transaction interacts with an XML repository containing one-to-one mappings between the components and their corresponding XML tags. Each component has at least two tags associated with it, namely a start-tag and a stop-tag. A first-in-last-out method is used to pre-affix and append start- and stop-tags to complex compositions of text and meta-data. These components can be reused either by including them in another LU, for example, or by referencing them.

5 EVALUATION OF PROTOTYPE

The prototype was made available to a group of seventy-five end-users belonging to the target user group identified earlier. The majority of this group were non-technologist teachers/instructors, and a smaller number included instructional designers with a technical background. Some members of this group were previously exposed to an authoring tool to code their learning units in EML (EML 2001, IMS 2002). This required them to work directly with the aforementioned XML editor. In general, this group showed excitement and satisfaction at using the prototype rather than working directly with XML/EML tags.

Feedback and evaluation tests showed that the users found our basic interface layout dealing with topics and events intuitive, and strongly related this layout to the paper-based preparation they were accustomed to. As anticipated, the users found the layout to be predictable and familiar. They also found it easy to generalise across different interface functions. Notable usability features embedded in the design of events include predictability and familiarity, as the names of the event types are exactly the terminology that instructors deal with every day. We found that the way the design is structured promoted author pre-emptiveness, generalisability and task migratability. We were largely satisfied that most of the learnability, flexibility and robustness parameters were addressed.

The following were identified as urgent requirements for the next version of the prototype:

- The inclusion of an author analysis function where the interface can sense the skills level of the author and adjust the display and functionality accordingly.
- The inclusion of a lesson preparation step enabling the author to enter initial ideas when planning a course, and also to be prompted with suitable strategies and hints.
- Special attention will have to be paid to the interface as the layout of this planning step can easily compromise the usability of the entire tool, should it require complex technical skills.
- Compliance with an open educational standard such as DMCI / SCORM (IEEE, n.d.), or full EML instead of a mere translation to XML tags.

6 CONCLUSION

Because an ASE network forms an intrinsic subset of a set of e-learning networks, existing ASE software requires expert users who are not challenged by their computing design environments, and are therefore able to focus their full attention on the design of a learning environment. However, lecturers are increasingly required to design e-learning environments, and as a result are challenged by the fact that their bags of professional skills do not, by default, include computing skills and natural software usage intuition. The implication of this challenge is that lecturers struggle to focus their attention on learning environment design, and are impeded by the design environment.

In this paper we proposed an approach to designing an ASE network that adheres to several of the important usability parameters known in software development, and at the same time produces reusable, XML-wrapped output learning objects. Our suggested approach of articulating required ASE elements as network nodes enables the designers to separate learning design issues from interface usability issues. The design of the prototype relied on the suggested methodology, where different events to be included in the ASE were designed through a set of formally defined nodes.

Our prototype proved to overcome many of the challenges that confront lecturers when they are designing e-learning software. The prototype is largely based on system pre-emptive dialogue initiative, which impedes flexibility to a certain extent. However, in the domain of authoring e-learning software, the most important usability design criteria are focussing on increasing usability for novice or occasional users, rather than expert users. A full implementation would obviously also consider the expert user, and include a more user pre-emptive approach. On the positive side, our prototype greatly enhanced learnability, robustness and most flexibility parameters.

The work reported upon in this article is partly based on work sponsored by a grant from the National Research Foundation of South Africa under Grant Number GUN: 2050310.

REFERENCES

Advanced Distributed Learning (ADL), (n.d). Retrieved April 2003 from http://www.adlnet.org.

ARIADNE, (n.d.). Retrieved April 2003 from http://www.adlnet.org.

ASTD Certification Institute, 2002. *ECC Standards*. Available from Retrieved October 2003 from http://www.astd.org/ecertification/standards.htm.

Badre A.N., 2002. *Shaping Web Usability*. Addison Wesley.

Cloete E. & Kotzé P., 2003. *Interaction parameters in the design of authoring support environments*. Technical Report: TR-UNISA-2003-01. Available from: http://www.cs.unisa.ac.za/TechnicalReports/

National Cancer Institute, (n.d.). *Usability Basics*. Retrieved October 2003 from http://usability.gov/basics/index.html

Cramer G., (n.d.) *Rapid Content Development Tools – Selection Criteria*. Retrieved April 2003 from http://www.radauthor.com/pdf/Tools%20%Selection%20Criteria.pdf.

De Vries, F., 2002. *Valkenburg group moving forward*. Retrieved October 2003 from http://learningnetworks.org/forums/

Dix A., Finley J., Abowd G.D., & Beale R., 2004. *Human Computer Interaction*. Prentice Hall, Third Edition.

Duchastel P., 2001. *Learnabililty.* Retrieved October 2003 from http://home.earthlink.net/~castelnet/info/learnability.htm.

EML, 2001. *Open Universiteit Nederland - Learning Networks.* Retrieved November 2002 from http://eml.ou.nl/eml-ou-nl.htm.

IDE, 2002. *Innovations in Distance Education.* Retrieved October 2003 from http://www.outreach.psu.edu/de/ide/.

IEEE P1484.12, (n.d). Learning object metadata working group (IEEE's LOM). Retrieved April 2003 from http://ltsc.ieee.org/wg12/index.html.

IMS, (2002). IMS Global Learning Consortium. Retrieved April 2003. Available at http://www.imsglobal.org.

Kotzé P., 1997. *The Use of Formal Methods in the Design of Interactive Authoring Support Environments*. PhD Thesis. Research Report YCST 97/09, Department of Computer Science, University of York (UK).

Macromedia Inc., 2002. *Macromedia Authorware Support Center*. Retrieved October 2003 from http://www.macromedia.com/support/authorware.

Mehlenbacher B., 2001. *Usable Web-based Instruction Resources*. Retrieved October 2003 from http://www4.ncsu.edu:8030/~brad_m/publications.html#Web-Based.

Miller M.G. & Cloete E., 2003. Authoring tool for technical and technically-challenged e-learning designers. In *Proceedings of CATE 2003, International Conference on Computers and Advanced Technology in Education.*

Preece J., Rogers Y., and Sharp H., 2002. *Interaction Design: Beyond Human Computer Interaction*. Addison Wesley.

Proctor R., 2002. *Usability I: Principles and guidelines*. Retrieved October 2003 from http://www.informatics.ed.ac.uk/teaching/modules/hci/slides6.pdf.

Step Two Designs Pty Ltd., 2002. How to evaluate a content management system. *KM Column*. Retrieved October 2003 from http://www.steptwo.com.au/papers/kmc_evaluate/index.html.

Tennyson, R.D., 2001. Defining core competencies of an instructional technologist. *Computers in Human Behavior*, 17(4), 355-362.

W3C, 2000. Authoring Tool Accessibility Guidelines 1.0. Retrieved October 2003 from http://www.w3.org/TR/ATAG10/.