Chapter number

Bringing Activity into E-Learning – the Development of Online Active Learning and Training Environments

Claus Pahl

Introduction

Interaction is central in learning processes (Moore, 1992; Ohl, 2001). E-learning systems can act as mediators in the interaction between the learner and her or his environment, i.e. content, peers, and instructors. Engaging the learner in a learning process through interaction is a central element of successful learning design (Sims, 1997). The current predominant focus on knowledge transfer in e-learning is partly a result of a lack of interactive multimedia technologies. With the recognition of skills training as being equally important to knowledge acquisition, more work has recently been done on activity-based learning and training supported by interactive multimedia technology (Trikic, 2001). With recent advances in multimedia and Web technologies, a shift from purely knowledge-based learning towards activity-based learning and training can be observed (Okamoto et al., 2001).

Interactive Web and multimedia technologies are enablers of support for skills-oriented training in learning technology systems in addition to knowledge-based learning. In particular in e-learning environments, the learner-content interaction is often more central than the learner's interaction with instructors and peers (Ohl, 2001). The focus here is on the development of active learning objects based on interactive educational multimedia. In a wide range of areas, training of activities and skills is of paramount importance. Advanced uses of multimedia, in particular interactive educational multimedia is a technical solution that can support activity-based e-learning and e-training (Elsom-Cook, 2001). Active learning is characterised by knowledge- or skills-level interaction of the learner with e-learning system and content, i.e. goes beyond the typical navigation and delivery management interaction. The input of the learner in interactions is meaningful in the context of the subject. Activities such as interactions between learner and instructor or collaborations between learners are neglected here.

Current approaches for knowledge transfer-oriented learning and their development do often not apply in this new context. Support frameworks for multimedia development for e-learning environments exist (Heller et al., 2001). However, the focus of these frameworks is mainly on knowledge acquisition. Understanding the requirements of activity-based learning and training and techniques to develop interactive educational multimedia are thus needed. Both developers (for instruction and software) and instructors need support for the development and deployment of this type of learning and supporting technology.

This chapter presents experience with the development of online active learning and training applications in a constructive form as a development framework. The aim is to guide the development of an activity-centred e-learning or e-training system for developers and instructors or the selection process of a suitable system needed by an instructor. The role of the instructor in the development process is explained. The focus is on interactive content development, based on a sound learning design. Those aspects that distinguish the development of media-based learning objects for active learning and training from classical knowledge-oriented learning and learning objects development are emphasised. The framework is illustrated using a third-level Web-based computing course in which an integration of various learning and training activities ranging from controlled animations to graphical modelling and text-based programming activities in a realistic setting is essential. This system – a database learning environment called the Interactive Database Learning Environment IDLE – is a typical example of a wide range of online learning systems that support activities through animations, simulations, and other forms of processing.

Literature Review

The focus of this chapter is on activity-based learning and training. Active learning plays an important role in recent instructional design approaches. Learning, however, is not an activity. It is a cognitive process that is rather a by-product of understanding (Mayes & Fowler, 1999). This cognitive dimension cannot be directly addressed using a software tool. Essential elements of the learning experience, like direct interactions between learner and instructor or other learners, are missing. Activity theory can explain the role of the e-learning system in this context. Learning can be seen as a process in which learners actively construct knowledge and acquire skills. In the context of computer-supported interactive learning and training environments, the role of this computer environment is that of a tool that mediates the interaction between learner and content. Activity theory is a conceptual framework that can describe the structure, development, and context of computer-supported learning activities (Nardi, 1997). The emphasis on the interaction between learners and their environment explains the principle of tool mediation. Tools shape the way humans interact with reality. Educational tools reflect experiences learners and instructors have made in trying to solve particular problems. The learner should be engaged in solving meaningful problems in an activity-based, realistic setting.

Learning should be an active process in which interactivity is central (Northrup, 2001). Moore (1992) distinguishes three types of interactivity – learner-learner, learner-instructor, and learner-content interactions. It is often argued (Sims, 1997; Ohl, 2001) that content has a more central function in computer-based education than interaction with peers or instructors. The focus here is on learner-content interactions, where content is provided in the form of interactive multimedia. Learner-content interaction in computer-supported learning and training actually occurs as interaction with the interactive multimedia features (Boyle, 1997). Educational multimedia systems are usually hypermedia systems providing structure through hierarchy and guidance for learning tasks through navigation topologies (Jonassen & Mandl, 1990). Crucial for educational multimedia are the multimedia interface and the interaction dialogues a multimedia system allows through channels (text, mouse, etc.) and languages (natural, formal, etc.).

The virtual apprenticeship model (Murray et al., 2003) is a pedagogical theory – based on the activity model – that defines an activity-based and skills-oriented learning and training framework for the IDLE system. An apprentice is a learner who is coached by a master to perform a specific task. In an e-learning and training environment, the master's role is often replaced by an intelligent software tool such as IDLE. The apprenticeship model determines a number of aspects including the activity purpose and the degree of involvement, interaction styles (e.g.

the organisation of learning into sessions and cycles), and the interconnectedness of activities and features. The virtual apprenticeship model puts an emphasis on skills-oriented activities with a high degree of involvement of the learner.

Usability is central in user-oriented systems such as interactive, multimedia-based e-learning systems that supports active learning. Technical and pedagogical usability can be distinguished as two aspects of usability (Melis et al., 2003). Technical usability aims at enabling trouble-free interaction of the user with the system. Pedagogical usability facilitates the learning process. In the context of e-learning systems, technical usability should enhance the pedagogical usability. Usability in both forms needs to be a central design and also evaluation focus for both developers and instructors.

The development of an e-learning system is a software development process that is heavily influenced by the specific context of the application domain, e-learning. Some process models to guide and structure the development process have already been proposed. Heller et al. (2001) have focused on the multimedia perspective, which is important for learning content development. Virvou and Tsiriga (2001) have used object-oriented software development, a very common software engineering methodology, to develop e-learning systems. Boyle (2003) follows a similar direction by combining object-oriented development methods with pedagogical principles for his development process model. None of the existing approaches, however, has addressed the specifics of active learning. In order to make a software process model accessible for domain experts, the process model also needs to be embedded into a learning-specific conceptual framework. The development of these systems requires the joint participation of software developers and instructors as domain expert in this process

Case Study

IDLE is the Interactive Database Learning Environment – a Web-based learning and training system for database modelling and programming. IDLE is based on the virtual apprenticeship model, which emphasises skills-oriented learning activities. The system acts as a mediator between a learner and interactive content in a realistic training environment. This case study aims to illustrate the importance of a systematic development approach for e-learning systems development and deployment and focuses on aspects that are often overlooked. Various forms of learning and training activities and how they can be supported by interactive multimedia are illustrated.

IDLE Overview

An interactive learning and training environment for SQL database programming is embedded into the online courseware system IDLE. Focussing on the SQL element illustrates the interactivity aspects. The SQL part forms a central part of this course as database programming is one of the core learning objectives of the course. Programming is a skill that needs to be trained by the learner. Moreover, this course is also an introduction to database engineering. Therefore, understanding and mastering the overall development process of a database application is equally important. Database programming in SQL also requires conceptual understanding of the underlying data model with its structures, operations, and constraints.

Solutions to programming problems, which are presented as a guided tour through the material, can be submitted through a Web interface to a remote database server, which executes the input and replies with data from a database, or error messages. Scaffolding in form of feedback, self-assessment functionality, and links to background material is available. The tutorial prepares the learner for coursework, such as lab tests and projects, and final exams. The IDLE system aims at providing the learner with a realistic learning context by integrating features and problems into a learning environment that are similar to tools and tasks that would be faced by a database engineer in a real development scenario.

Classification of IDLE Features

Four different features for the different aspects of database programming are provided:

- Conceptual knowledge. Conceptual knowledge is presented in a virtual lecture system. The learners use an audio-visual presentation that presents the conceptual background. Recorded speech of the lecturer is synchronised with material in overhead form. Learners can control this lecture-style presentation through the usual interactive features of an audio player.

- Procedural knowledge. SQL is about the execution of instructions. Procedural knowledge is presented in an animated tutorial system. SQL is a language that implements different database operations. An animated tutorial using flash animations illustrates the execution of these operations is a step-by-step fashion using examples. Learners can execute these operations in small steps. Animated tutorials are particularly useful to illustrate operations that are sometimes conceptually difficult to understand.

- Programming skills. SQL programming is the core activity, supported by an interactive tutorial that guides the learner through exercises to be worked on within the system. SQL queries are often complex and difficult to formulate. Queries are supported through an interactive tutorial. The tutorial guides a learner through a sequence of exercises with increasing difficulty. Each unit addresses a particular aspect of SQL querying. The feature provides an input interface for each exercise where a learner can type in an SQL solution and submit this solution to a remote database server that executes the query and returns the result. Syntactic and semantic feedback is available. This feature provides links to the relevant background material (conceptual and procedural knowledge features).

- Development skills. SQL programming is part of the overall database application development process, which supported by an integrated lab environment with modelling, programming, and analysis features. The development of a database application is a multi-stage process including the stages modelling, programming, and analysis and optimisation. The database course environment provides interactive, integrated lab features for all three activities. The learner is provided with a workspace in which s/he can create and store a data model, which is interconnected with the other features. An integrated, realistic lab environment that resembles features of database development environments is the central feature.

Conceptual understanding of principles and concepts is of course required before practical work can start. However, the aim of the tutorial system is to allow learners to go quickly into the practical features by supporting a learning-by-discovery style, allowing them to acquire skills, but also to construct and deepen their conceptual knowledge through activities in meaningful and realistic problems. Consequently, the practical features are well linked to the respective background.

An Interaction Space – Activity and Interaction in Learning and Training

This design section introduces an activity and interaction design framework – called an interaction space – focussing on learning activities and training. Mapping learning and activity types onto the most suitable media type is an essential step that shall be illustrated by the IDLE implementation. Central is a classification of learner activities based on various aspects such as learning purpose (knowledge or skills) or degree of involvement. The classification is illustrated by different activity types supported within IDLE. This section focuses on early design stages, drawing attention to some aspects that are sometimes ignored, before classical learning design instruments would be deployed. This is a stage where in particular instructors are highly involved as domain experts.

Active Learning

In IDLE, database application development provides a meaningful problem that requires a learner to develop and deploy a database application with its structural and operational elements within the learning environment. The database courseware system shall create a realistic setting by integrating tools into a learning and training environment that resemble tools of a real database development environment. Modelling, programming, and analysis features interacting with an enterprise-scale database server can provide a realistic setting. IDLE is a software tool that facilitates these activities in a guided learning process. Learners learn to solve problems in a dialogue with the system.

The following instructional guidelines, developed by domain experts, have served as requirements for the IDLE system development in order to enable active learning and training in a pedagogical framework defined by activity theory:

- The active participation of the learner is essential.
- Active construction of knowledge and skills results in an increased ownership of the learner in the learning process.

- Meaningful projects allow learners to acquire skills and experience in database programming and development.
- A realistic setting improves the learning experience and demonstrates the applicability of knowledge and skills.
- Guidance and feedback provide instructional support in the environment.

Activities such as programming are at the centre of the IDLE learning and training strategy. However, supporting the learner through scaffolding, e.g. guidance and feedback, is equally essential from the instructional perspective (McLoughlin et.al., 2000). In addition to mediating between learner and database tools, the environment must fulfil functions of the instructor. The environment needs to replace central tasks of the instructor in form of a virtual master that guides a learner through exercises and that provides immediate feedback on activities. Again, instructors can provide crucial input here. Each learning and training activity needs to be complemented by links to the background (conceptual and procedural knowledge in form of virtual lectures and animated tutorials) relevant and problem-related for the activity in question.

Interaction Model

The learning and training activities facilitated by educational multimedia interactions between learner and content shall be captured in form of an interaction model for learning activities. The aim of this interaction model is to support the design of educationally sound interactive learning activities. A taxonomy of activity types is a central element of this interaction model that helps to categorise activities at an early stage. The categorisation provides the developer and instructor with pointers to best practice, since each category is usually implemented in a specific way. Two aspects of activity types – purpose and involvement – that help to describe activities can be distinguished.

Three activity types can be defined based on the purpose of the learning process:

- Declarative knowledge acquisition activities: the aim is the acquisition of declarative knowledge in order to reason about it.

- Procedural knowledge acquisition activities: the aim is the acquisition of procedural knowledge in order to reason about it.

- Skills acquisition activities: the aim is the acquisition of procedural knowledge and experience in order to perform the instructions.

The second category is important in particular in the sciences and engineering domain where an understanding of the subject activities is required for a learner.

The style of the activity execution can be characterised based on the degree of involvement and influence of the learner on the environment. Types ranging from system-controlled to learner-controlled environments can be distinguished:

- Observation: a form of knowledge acquisition with no influence on the environment activities by a passive learner.

- Controlling: a form of knowledge acquisition mixed with knowledge production, based on observational elements, but allowing the learner to influence the environment activities to control their ordering.

- Creation: a form of activity where knowledge or skills are created by producing some form of artefact that can be processed by the learning environment.

The individual types for each of the categories are not meant to be exclusive – a more fine-granular classification can replace the above types if needed. Often the two aspects of activity types are related. Declarative knowledge is often acquired through observation, procedural knowledge for reasoning purposes through controlled animations, and skills through artefact creation and processing. The learning-by-doing idea is part of the IDLE active learning approach. It captures the interplay of knowledge acquisition and knowledge creation in an interactive process with the learning environment. This focus is widened in IDLE by considering knowledge acquisition on the one hand and skills and experience acquisition on the other hand as dual sides of learning and training.

In an interaction model, the activity model is the central element that defines content interactions, but which is complemented by two other aspects: goals, which are meta-level descriptions that define the learning objectives of an activity, and knowledge objects, which capture the knowledge underlying each activity in declarative and procedural form.

The ultimate objective at this development stage is the identification and the design of interactive learning objects that implement the activities in question and allow the learner to achieve the goals defined (Boyle, 2003). The interaction model – consisting of activities, their goals, and the underlying knowledge objects – helps the instructor and content developer to gather the required information.

Choosing the virtual apprenticeship approach as the underlying pedagogical framework is the first main IDLE design decision, which defines the context for the interaction model. Based on this decision, the characterisation of individual activities, like the SQL programming described above, can start.

The main IDLE activity categories in terms of the interaction model are summarised in Table 1. Further categorisation is, however, necessary for a detailed design. For instance, the lab activity could be refined into specific activities such as graphical design, programming, or optimisation. In terms of learning objects – learning object identification is the central objective at this stage – the aim is representing each activity in Table 1 by a composite learning object. This example illustrates the need to involve instructors in this process. Within each of these composite learning objects, individual object will address specific topics. One of these specific activities shall be illustrated now, looking in particular at the interactions between learner and learning objects and at the composition (sequencing) of smaller activities.

Activity	*Activity Type (Purpose)*	*Activity Type (Involvement)*
lecture participation	declarative knowledge acquisition	observation
tutorial participation	procedural knowledge acquisition	controlling
lab participation	skills acquisition	creation

Table 1. Some IDLE Activities and their Types based on Learning Purpose and Degree of Involvement

One of the skills acquisition activities in the IDLE system is SQL database programming, which has the goal of equipping learners with the ability to formulate complex database queries. Integrated with a database system, the learner – a virtual apprentice – works through guided material covering a range of individual problems. Each problem is based on a submission- and execution-cycle with a high degree of involvement of the learner through knowledge creation. Each solution – content-specific knowledge that is created by the learner – is analysed and, based on an individual activity history and integrated assessments, personalised feedback is given by a virtual master. At this level, the concern is the abstract classification of learner activities from the interaction model in order to design the learning object. For the database course IDLE the central design decision at this level is to focus on an integrated approach with a strong support of skills training activities. Detailed design could be supported by classical storyboarding approaches, which would develop generalised learning scenarios that can be used as a communications medium between software, content, and instruction developer.

An Interaction Infrastructure – Interactive Educational Multimedia

This implementation section discusses the technical aspects of interactive educational multimedia in the context of interaction for activity-based learning and training. Technical notions such as channels and interface languages shall be introduced as central aspects of educational multimedia that explain interactions between learners and content. Designing active learning and training in terms of abstract media concepts is central for the correct and successful implementation of the learning design from previous stages.

Interactive Multimedia

Interactive multimedia for activity-based learning and training can be distinguished into interaction with knowledge media and with activity media. Activity-based training focuses on skills-oriented activities, but needs to be integrated with knowledge learning aspects. Knowledge media focus on knowledge information to be communicated. Activity media focus on knowledge-based artefacts that are produced and processed in activities. The purpose of interactive educational multimedia is twofold:

- In addition to knowledge-level interaction, domain-specific activities need to be facilitated, i.e. activity-level interaction with educational multimedia feature through artefacts and instructions has to be enabled.

- The instructor can be replaced by a virtual form of an intelligent educational multimedia feature that provides advice and feedback, thus adding more meaning to the interaction.

Educational multimedia can be classified through different metadata facets:

- Channels are abstractions of a communication device, characterised by modality.

- Languages enable the encoding of information in a common notation for the communication over a channel.

- The activity purpose distinguishes whether declarative knowledge reasoning, procedural knowledge reasoning, or skills acquisition is aimed at.

- The activity style allows the classification of activities into observation, controlling, and creation, which describes the degree of influence of a learner on the environment.

- The content topic is the topic or domain within which activities or knowledge-level access is provided.

Learning objects have been used as a design notion in the previous section. These learning object designs and their underlying activities need to be implemented, which is mainly the work of a software or content developer. The focus here is on learning objects that realise learning activities as learner-content interactions. In particular the media to implement activities are of central importance. The term interactive educational multimedia (IEMM) shall denote media types especially suited to implement active learning objects. A number of standard mappings between activities and media types have emerged:

- Lecture participation (aiming at declarative knowledge acquisition) is often suitably implemented by audio-visual presentations.

- Tutorial participation (aiming at procedural knowledge acquisition) is often suitably implemented by animations.

- Lab participation (aiming at skills acquisition) is often suitably implemented by simulation or execution of activities.

Specific media types allow for the appropriate level and degree of interaction. Determining the most suitable media type and the interaction with the medium is the central activity at this stage. The following classification scheme and the investigation into interaction channels in the remainder of this section support this process to see a learning object as an interactive multimedia object and it helps to map designs to implementations.

IDLE supports three classical forms of third-level teaching – lectures, tutorials, and labs – in a virtual format. These three forms can be described using the facets of the educational multimedia classification scheme – see Table 2, which shows how some selected learning activity styles for particular topics are mapped onto multimedia features. For example, a simulation can be a subcategory of a moving pictures/images language. However, the elements of simulations can be identified and have meaning in the context of content (e.g. tables or records in the database context). Equally, operations (simulation activities) are represented as procedural knowledge.

Facet *Activity*	*Channel*	*Language*	*Purpose*	*Type*	*Knowledge Object*
lecture	text and audio	natural language	declarative knowledge	observation	introduction to databases
tutorial	dynamic animation	simulation	procedural knowledge	controlling	relational algebra
lab	text	formal language	skills training	creation	SQL

Table 2. Sample IDLE Media Classification

Interaction Channels

Multimedia is about channels and meaningful communication along these channels. Often, a natural language such as English is used over a text channel (written English) or over an audio channel (spoken English). For this context, a number of education-specific channels can be identified – supporting partly more formal languages, partly languages specific to the subject or instruction context. Two types of channels can be distinguished – those that support core content-oriented learning activities and those that are part of the meta-context of instruction; the latter including instruction-related learner actions and coaching actions by a master or instructor.

- Declarative knowledge (supporting core activity): declarative knowledge usually communicated in a domain-specific natural or formal language.
- Procedural knowledge (supporting core activity): procedural knowledge usually communicated in a domain-specific natural or formal language.
- Skills (supporting core activity): artefacts to be processed in form of activities are communicated with corresponding execution instructions.
- Actions (supporting meta activity): instruction-related actions executed by the learner, typically navigation or location of learning objects.
- Feedback (supporting meta activity): response of the system for each core activity.
- Coaching (supporting meta activity): meta-level information capturing an instructor's advice and guidance.

Multimedia interface languages capture and constrain the channel communications. A language defines the interaction dialogues; it describes the legal actions, how a learner can engage in an activity or how a learner can perform a task towards a learning goal.

The IDLE channel and language characterisation in Table 2 is high-level. These two aspects can be described in more detail. Table 3 provides a channel-oriented view on IDLE; it lists the educational channel types and some sample features that are based on these channels.

Channel	*Feature*	*Activity*	*Language*
declarative knowledge	database introduction lecture	HMTL and audio-based synchronised virtual lecture	natural language (written and spoken)

procedural knowledge	relational algebra animation	interactive simulation of algebra operator execution	formal language (interaction – animation control)
skills	SQL programming lab	submission of query solutions and dynamic page update by system	formal language – SQL (solution and result)
action	SQL tutorial navigation	guided tour through a series of connected exercises	formal language (interaction – navigation)
feedback	SQL programming lab	correction and provision of partial solutions for exercises	semi-formal language (error classification)
coaching	self-assessment	multiple choice questions and virtual master's feedback	natural language (written)

Table 3. Sample IDLE Media Channels

Interaction Specification

Languages for the educational context can be classified based on content-related aspects:

- Natural languages – in text or audio form – are often the basis of content.
- Formal languages – in text form – are often involved if some form of mechanical computerised processing is part of the subject domain.
- Simulations – automated execution of some real-world activities – are based on objects and procedures from the subject domain.

Besides the content aspect, dialogue and interaction patterns form the instructional aspect addressed by interface languages. On the most basic level the learner interacts with multimedia usually through keyboard- and mouse-based input; output can be static visual (text, graphics), dynamic visual (animations, video), or involving other modalities such as audio. The basic inputs are part of low-level activities such as navigation (knowledge acquisition request) or text input/submission (knowledge generation). A learning activity can be composed of more basic activities. The dialogue part of an interaction language consists of

- basic activities like select (knowledge acquisition by learner), submit (knowledge generation by learner), reply (response to knowledge acquisition/generation);
- activity combinators like sequence, iteration, choice;
- system components such as learner and multimedia learning objects in an e-learning architecture.

An example shall illustrate how this language is used to express abstract interaction behaviour between learners and learning objects. The expression

iterate

sequence of

LR.*select(exercise)*
LR.*submit(solution)*
LO.*reply(result)*

is the interaction specification of an exercise activity scenario. A language needs to facilitate declarative and procedural knowledge communication, skills-oriented activity execution, learner actions, and meta-level pedagogical interactions (coaching). The select activity denotes a learner action; submit and reply support skills-oriented activities; reply could, in addition to results for e.g. SQL submissions, also convey meta-level feedback and coaching. This specification distinguishes between learner (LR) and multimedia learning object (LO). For instance, the SQL multimedia lab system replies with a result that includes the result of the execution of the previously submitted solution and additional feedback for the learner.

Interaction is actually central in two dimensions, of which one only has been looked at so far:

- Interaction between learner and learning object – which has just been looked at – is based on multimedia interactions. A simple notation for the abstract interaction specification was illustrated, which can serve as a basis for the media implementation. Here platform and media-type specifics need to be applied. For instance, the Web provides the required submission and reply mechanism in form of HTML forms and HTTP protocol operations.

- Interaction between learning objects – both intra-learning object and inter-learning object sequencing – is based on hypermedia navigation. The learner interacts with the system by navigation through the multimedia learning objects based on an educationally sound navigation infrastructure defined by the instructor.

The second category is usually supported by platform-specific languages and implemented in form of HTML links. Although for instance SCORM sequencing (ADL, 2004), one of the most widely accepted standard for learning technology, is a language to define instructional sequences, platform-specific notations are ignored here, as these are not specific to activity-based learning and training.

Case Study Evaluation

Case Study Deployment

The case study system is actually in use since 1999 to support a database course in an undergraduate computing degree and has been extended since then substantially. The IDLE system has been continuously evaluated since 1999. More than 1000 students from computing, mathematics, and engineering degree programmes have used the system so far. IDLE is an e-learning system supporting an on-campus course. In order to understand whether the proposed development framework actually results in an effective e-learning solution can only be decided by carrying out a thorough evaluation of the developed system as the product and also the development process. While software and multimedia aspects such as maintainability and cost-effectiveness are important (and will be discussed), the educational perspective focussing on learner behaviour, effectiveness, and the learning experience, which are ultimately the decisive criteria, shall be looked at first.

Evaluation Approach

Whether learners actually use an e-learning and training environment as expected by the instructor is often a question that is difficult to answer. Learner behaviour in tool-mediated environments is determined by the learners' motivation, their acceptance of the pedagogical approach and the technical environment, their learning organisation, and their activities in the environment. This evaluation should extract factors that make learning successful for the learner and behaviour accountable and predictable for the instructor. Success shall be defined in terms of the learner's acceptance (reflected directly though survey data and indirectly through usage data) and effectiveness in terms of high levels of attainment. Meta-level indicators such as motivation and acceptance and analyses based on interaction patterns such as learning organisation and usage are distinguished as the main aspects of the evaluation.

The behaviour of learners in e-learning environments is influenced by the meta-level indicators 'motivation to use the system' and 'acceptance of the approach'.

- Motivation, the reason to do something, causes the learner to act in some planned and organised way, giving the activities a purpose. It explains when and what purpose learning and training resources are used for.
- Acceptance, i.e. to follow the learning approach and use the system willingly, is crucial for the introduction of new educational technology. It can be explained in terms of surveys and observations about frequency and regularity of usage.

These two factors are usually not the decisive ones, but they provide nonetheless valuable insights and allow the interpretation of other observations.

Two aspects of the learners' interaction behaviour within the learning activity can be distinguished. Firstly, the learning organisation addresses the study habits and captures how learners organise their studies over a longer period of time. Secondly, the usage of the system captures single learning activities and embraces how the learner works with the system in a single study session.

- Organisation – the way activities are planned and put into logical order – reflects study habits and is guided by the learning purpose. It looks at large-scale interaction patterns.
- Usage – the way the system is used – reflects the actual learning activities of the learner. It looks at small-scale interaction patterns.

Only abstracted patterns of interaction provide an insight into general system usage.

Learner Behaviour and Effectiveness

The instruments for the behaviour analysis include survey methods to address motivation and acceptance and Web usage mining techniques to capture organisation and usage in a Web environment (Pahl, 2004). This combination of methods provides a more complete and accurate picture than uses of survey and learner observation or learner tracking features available in various teaching and learning platforms.

- Motivation. There is a very clear preference for coursework preparation as the main motivation. A Web log analysis shows that the SQL lab feature is mainly used to support coursework and to a lesser extent for exam preparation, which confirms the survey result. More insight into the motivation of the learner's study organisation comes from a question regarding the main values of a virtual system – 'always available' and 'self-paced learning' are seen as the key advantages.
- Acceptance. Learners have indicated an overall acceptance of tool-mediated active learning as the pedagogical approach. Comparing traditional and virtual tutorials gives a more differentiated view on acceptance. No favourite emerges; this clearly demonstrates that learners accept virtual tutorials as equally suitable and effective. Another indicator for the acceptance of the approach of self-directed active learning is reflected by frequent and regular usage of the system. While usage mining shows that the tutorial system has not been used frequently and regularly over the whole term, it has however been used intensively in certain periods to fulfil a particular purpose.
- Organisation. The study organisation – the self-paced learning aspect – shows a just-in-time learning approach of learners with high usage immediately before coursework deadlines and examinations. Web usage mining can give us a clearer picture about the organisation. Session classification allows us to determine the purpose of learning sessions, for instance, attending virtual lectures or practising in virtual tutorials, and to compare the session purposes of different periods. Interactive services are heavily used during term, but less so for the exam preparation.
- Usage. Besides the long-term study organisation, analysing learning activities within a study session is crucial in order to understand

how learners learn. An abstract picture of the purpose(s) of each session can be provided by a session classification technique, but a detailed look at how learners interact with the system, whether they repeat units, or whether they combine interactive elements with lectures is also necessary. Extracting these interaction patterns is highly important and it is paramount for the instructional designer to compare actual and expected behaviour. The interaction design and its implementation reflect expected, but needs to facilitate the learner's preferred learning behaviour.

Learning behaviour analysis is an essential instrument for effectiveness evaluations of tool-mediated active learning. However, effectiveness also comprises aspects such as learner attainment. Exam and coursework results – in comparison with a traditional delivery – can prove the effectiveness with respect to attainment, which is one of the critical success factors. In the system, the learner attainment level of the traditional delivery has always been reached since the IDLE system was introduced in 1999. A minimal correlation exists between usage and attainment for the SQL tutorial and lab system, which can be interpreted as an indication for the benefit of using the interactive tutorial and lab features for exam preparation and coursework. In contrast to virtual lecture attendance, where books can serve as an alternative, interactive tools to acquire programming skills are more difficult to replace and, therefore, their usage is beneficial. The high number of students actually using the non-compulsory IDLE system proves this point. Only less than 10% of each class decide not to use the system.

Integrated Evaluation and Design Process

An iterative development approach based on formative evaluations that are fed back into design and implementation is necessary to adjust learning processes to the new active media environment. A number of examples shall illustrate this point:

- Activity characterisations in terms of purpose and involvement can be looked at using the behaviour analyses (frequency and regularity) and survey results to determine the learners' acceptance of a specific activity type.
- The adequacy of learning goals based on underlying knowledge aspects can be established based on attainment evaluation, but also

by classifying learning sessions based on potential learner objectives.

- The specification of the learner-content interaction (which is expected and supported behaviour) can be validated using an interaction pattern analysis.
- The suitability of multimedia and channel types can be verified by looking at broader frequency and regularity patterns.

The evaluations help us to scrutinise essential design and implementation decisions. Evaluations also address unexpected aspects, such as just-in-time learning patterns, that would have to be readdressed in revisions of the design.

Software Development Issues

Usability is an important software quality. The literature distinguishes two aspects of usability – technical and pedagogical usability. The interaction infrastructure is the technical side that addresses technical usability, whereas the interaction space based on the interaction model needs to be based on pedagogical usability aspects. Technical aspects such as media types and interaction designs are fitted into a pedagogical framework of learning activity types in order to allow the interaction infrastructure to support learning interactions defined in the interaction space. Classical Web usability issues, such as page layout, are neglected here, allowing more emphasis to be put on pedagogical issues such as motivation, acceptance, organisation, and usage.

This chapter has suggested some development techniques for e-learning systems. Although the focus here was on the implementation of active learning, addressing software technologies aspects should not be neglected. One central lesson learned – supported by other authors, e.g. Palmer and Tulloch (2001) – concerns software properties. Often e-learning systems are research prototypes and showpieces. These systems have turned out to be costly and difficult to maintain, if used on a day-to-day basis. The development of easy-to-use infrastructures and mainstream system designs should be favoured if maintainability of learning objects and e-learning applications and not the exploration of new technologies is the objective. Even in the latter case, the technologies might remain in service for several years and, consequently, need to be maintained as well. These observations should always be considered in the design of learning objects when choices regarding media types or infrastructure technologies are made.

Discussion

Although the overall experience with the case study system is positive, some problems have occurred. Therefore, important limitations and weaknesses for learners, instructors, and developers shall be discussed.

- The system replaces the direct interaction between learner and instructor to some extent. The level and quality of feedback for the learner is consequently reduced – although it is worth noting that the learner can get more feedback at a lower level due to the constant availability of the system. The instructor equally receives less direct feedback and has more difficulty in monitoring the course activity.
- The analysis of usage patterns and learning organisation shows a just-in-time learning behaviour. Although instructors prefer a more regular approach to learning, this reflects common learner behaviour. The motivation is a central factor in what determines a successful learning experience.
- The cost of developing and deploying an e-learning system such as the presented one is often prohibitive for a single organisation or provider. Only sharing and the reuse can provide a solution in many cases.

These more practical limitations are completed by a more foundational limitation that was already discussed in the literature review. Learning is not an activity and cannot be supported directly. The system discussed here can virtualise tools and their environment in some contexts well, but the instructor is more difficult to replace. Only some basic instructor tasks have been attempted in order to relieve the instructor from these, such as feedback and personalised guidance in a range of standard situations.

A central question that needs to be discussed is the transferability of the approach to other topics and other groups of learners. In the context of active learning and training support, similar technologies have been used for other subjects beyond the computing domain. In engineering and sciences, so-called virtual instruments are often used to enhance training aspects. Learners can for example carry out mechanics or physics experiments and can analyse the resulting data using these virtual instruments, which usually combine a multimedia representation with an analytical tool. Another example of e-learning systems for active learning and training are simulators used to train people in the usage and handling of complex systems ranging from aircraft to powerplants to medical devices. These combine multimedia content with real-time computation features. The pedagogy does not necessarily need to be changed, as the principle is still a learning experience based on interactive exercises and problems. Only the degree of personalisation might vary from application to application.

The presented technology is transferable to other groups of students. The case study has been used by students across a number of backgrounds, though predominantly from computing. With increasing computer literacy of younger learners and even advanced IT skills of third-level students in technical disciplines, these media- and interaction-intensive e-learning systems are common. The case study system is accessible using Web browser technology and does not exceed the complexity of standard PC desktop tools. making it in principle accessible for a broad range of learners.

Conclusions

Essential problems that a content developer faces in the development of learning objects for active learning and training relate to very practical aspects. What media types work for a particular learning activity or at what cost can these learning objects be developed, deployed, and maintained are central design issues. A development approach for active learning objects has been presented that provides a framework to answer some of these questions and that highlights the difficulties specific to active learning and its implementation through interactive educational multimedia.

Experience shows the importance of knowledge-or skills-level interaction, i.e. interaction that is meaningful within the context of the subject domain (Ravenscroft et.al., 1998). The interaction model that has been presented is an important feature that captures central elements of a good interaction design by linking learning interaction (the educational perspective – the interaction space) with multimedia interaction (the technical perspective – the interaction infrastructure). In addition to the interaction model that structures an interaction space, an interaction infrastructure is needed. The interaction infrastructure enables learners to achieve their learning objectives in the system. The interaction infrastructure consists of media to convey content, a navigation topology for sequencing, and learner-media interactions for knowledge- and skills-level activities. The evaluation of the approach has looked at meta-level indicators and usage observations – both are important to decide whether the learning objectives can be realised and whether the learning object implementation is effective and successful.

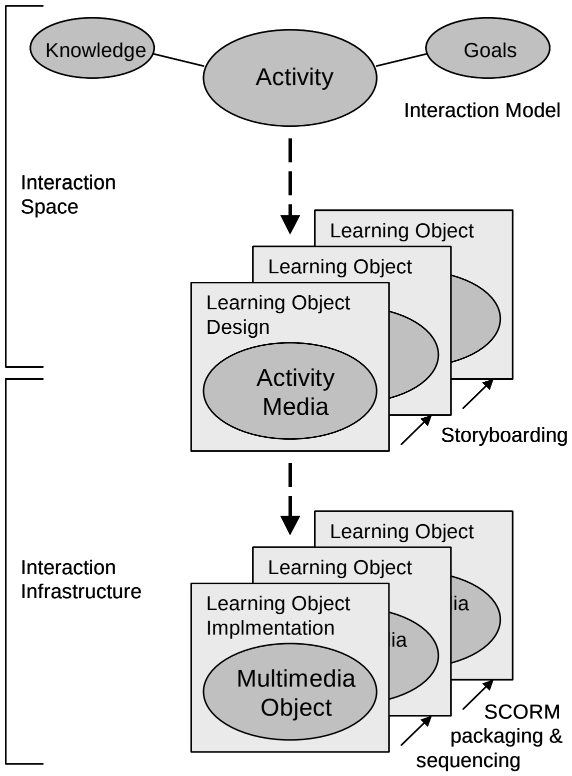

Fig. 1 Active Learning Object Development Process.

The development approach influences both the educational quality and the software quality of an e-learning system. The method is feasible, allowing the implementation of an effective e-learning solution. The evaluation, however, has also highlighted critical aspects:

- On the educational side, learning behaviour is difficult to predict and model. Only an iterative approach involving design, implementation, and formative evaluation provides a solution. Instructional design is difficult in novel, technology-supported environments. Designing instruction such that the learner's behaviour corresponds to the instructor's expectations is a challenge.
- On the technical side, the costs of maintenance and change are often underestimated, resulting in these aspects being ignored in learning object design. Change should be considered in the design from the outset. This is in particular the case, if advanced technological solutions based on interactive educational multimedia are involved.

Some observations show the link between interactive educational multimedia design and our formative evaluation results. The more interactive the features, the more they are used. The SQL feature is a good example of this, which supports the hypothesis of active learning as an effective form of instruction. The media types and navigation infrastructure to support interactivity not only on a technical level, but also in an educationally sound an effective way, need to be scrutinised through evaluation as our experience with initially unexpected and undesired behaviour suggests.

One of the lessons learned from the case study is that a development and management methodology for e-learning technology is needed. The objective here has been to raise an awareness of the solutions, but also the technical problems in the development of e-learning solutions. Although the discussion has focussed on a computing subject, the approach is not specific to computing, and can be applied to any other domain. The media types addressed are standard media supported by Web infrastructure. What might vary from subject to subject is the preferred media type, whether it is text-based if more formal languages are involved or whether more graphical types prevail. Overall, this chapter has pointed out some aspects of active learning object development, see Fig. 1 – a full design methodology would have been beyond the scope of this investigation.

References

ADL. (2004). *Sharable Content Object Reference Model* (SCORM) – Content Aggregation Model (CAM), Run-Time Environment (RTE), Sequencing and Navigation (SN). http://www.adlnet.org/index.cfm?fuseaction=scormabt. (visited 06/04/2006).

Boyle, T. (1997). *Design for Multimedia Learning*, Prentice Hall, Europe.

Boyle, T. (2003). Design principles for authoring dynamic, reusable learning objects. *Australian Journal of Educational Technology*, 19(1), 46-58.

Elsom-Cook, M. (2001). *Principles of Interactive Multimedia*. London: McGraw-Hill.

Fowler, C.J.H., & Mayes, J.T. (1999). Learning relationships: from theory to design. *Association for Learning Technology Journal*, 7(3):6-16.

Heller, R.A., Martin, C.D., Haneef, N. & Gievska-Krliu, S. (2001). Using a Theoretical Multimedia Taxonomy Framework. *Journal of Educational Resources in Computing*, 1(1).

Jonassen, D.H. & Mandl, H. (Eds). (1990). *Designing Hypermedia for Learning*. Berlin: Springer-Verlag.

McLoughlin, C., Winnips, J. C., and Oliver R., (2000). Supporting Constructivist Learning through Learner Support On-line. In *Proceedings of the World Conference on Educational Multimedia, Hypermedia and Telecommunications EDMEDIA 2000*, pp 674-680.

Melis, E., Weber, M. & Andres, E. (2003). Lessons for (pedagogic) usability of elearning systems. *Proceedings Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education eLearn-2003*, pp 281-284.

Moore, M.G. (1992). Three types of interaction. *The American Journal of Distance Education*, 3(2), 1-6.

Murray, S., Ryan, J. & Pahl, C. (2003). A tool-mediated cognitive apprenticeship approach for a computer engineering course. *IEEE International Conference on Advanced Learning Technologies ICALT'03*, 2-6.

Nardi, B. (ed.) (1997). *Educational Context and Consciousness: Activity Theory and Human-Computer Interaction*. MIT Press.

Northrup, P. (2001). A framework for designing interactivity into Web-based instruction. *Educational Technology*, 41(2), 31-39.

Ohl, T.M. (2001). An interaction-centric learning model. *Journal of Educational Multimedia and Hypermedia*, 10(4), 311-332.

Okamoto, T., Cristea, A. & Kayama, M. (2001). Future integrated learning environments with multimedia. *Journal of Computer Assisted Learning*, 17, 4-12.

Pahl, C. (2004). Data Mining Technology for the Evaluation of Learning Content Interaction. *International Journal on E-Learning*, 3(4), 48-59.

Ravenscroft, A., Tait, K., and Hughes, I. (1998). Beyond the Media: Knowledge Level Interaction and Guided Integration for CBL Systems. *Computers & Education*, 30(1/2), 49-56.

Sims, R. (1997). Interactive learning as "emerging" technology: a reassessment of interactive and instructional design strategies. *Australian Journal of Educational Technology*, 13(1), 68-84.

Trikic, A. (2001). Evolving open learning environments using hypermedia technology. *Journal of Computer Assisted Learning*, 17, 186-199.

Virvou, M. & Tsiriga, V. (2001). An Object-Oriented Software Life Cycle of an Intelligent Tutoring System. *Journal of Computer Assisted Learning*, 17:200-205

Biography

Dr. Claus Pahl is a Senior Lecturer and the leader of the Web and Software Engineering research group at Dublin City University, which focuses on Web technologies and e-learning applications in particular. Claus has published more than 120 papers including a wide range of journal articles, book chapters, and conference contributions on e-learning. He is on the editorial board of the International Journal on E-Learning and is a regular reviewer for journals and conferences in the area of software, Web, and learning technologies and their applications. He has extensive experience in educational technologies, both as an instructor using technology-supported teaching and learning at undergraduate and postgraduate level and as a researcher in Web-based learning technology. The IDLE environment, developed by him and his students, is in use in undergraduate teaching since 1999.