Educational Content Creation and Sharing in a Technology-rich Environment

Bernd J. Krämer and Peng Han
FernUniversiät in Hagen
Department of Mathematics and Computer Science
58084 Hagen, Germany
Email: {han.peng|bernd.kraemer}@fernuni-hagen.de

***Abstract*— The componentization and reuse of topical information and the organization of learning processes according to pedagogical theories have long been discussed separately in e-learning literature. The former was led under the buzzword "learning object", the latter focussed on learning activity sequencing and culminated in the Learning Design standard. This paper sketches a methodological framework and an e-learning portal that reconcile both strands of discussion in a comprehensive support in digital learning content production, adaptation and reuse. It presents an approach towards developing and reusing interactive learning objects relying on software design principles and adaptation mechanisms such as late composition and parameterization. Topical information like facts, concepts, procedures, processes or principles of a knowledge domain can be flexibly combined with learning objectives and activities supporting the learning process of an individual or group of learners. It suggests keeping information and educational context separate at design time and connecting both facets of learning objects only at reuse time. Parameterization is a software design principle used here to facilitate the adaptation of a learning object to different themes and didactic contexts. These design principles are illustrated for Java applets and for interactive Flash animations. This paper also illustrates facilities to adapt predefined didactic scenario templates, design new scenarios and update them with reusable learning objects from a repository or from the author's workspace.**

***Index Terms*— Learning object; configurable learning object; cognitive taxonomy; didactic scenario, didactic parameterization; content reuse; IPR; licensing.**

I. Introduction

The concept of learning objects arose in the early nineties driven by the motivation to reduce the development and maintenance cost of digital learning resources through modularization and reuse. Learning objects promised to offer a new way to create and mediate educational content in terms of smaller units of learning. These units are self-contained, can be reused in multiple contexts and different educational settings, and can be combined into coherent collections of learning materials. If didactically well designed, interactive learning objects can help students to understand comprehensive concepts and the inner working of complex processes better than from mere textual descriptions and static figures. This is particularly acute in self-paced learning situations, in which interactive multimedia learning objects can stimulate higher-level cognitive skills by allowing students to carry out procedures, to organize components of concepts or virtual materials, or even create new solutions.

The design and implementation of interactive learning objects is, however, time-consuming and requires special skills. Learning objects are also typically localized and tightly connected with particular didactic scenarios. However, this strongly limits their reuse in different contexts.

A. The CampusContent Project

In the main body of this paper, we present some findings made and results produced in the CampusContent project. CampusContent[1] is a competence center for e-learning that has been funded between March 2005 and July 2009 by the Deutsche Forschungsgemeinschaft[2]. The project was motivated by the observation that although a huge number of digital learning materials has been developed in the last decade, availability and access to these resources are limited and the degree of reuse is disappointingly low. The German Federal Minister of Education and Research, for example, invested millions of Euro at the beginning of this century in a four-year funding program called "New Media in Education", which aimed at the production of high quality digital learning content in and for German universities. However, the plethora of educational content resulting from such projects is difficult to find; it is not sustainably managed, and rarely has been designed for adaption and reuse in different learning arrangements. In contrast to books and scholarly journals that are systematically catalogued, managed and cross-referenced by libraries, no widely accepted archiving system and indexing standard exist that enable the systematic and effective storing, acquisition, distribution, and easy exchange of digital learning materials and representations of successful applications of didactic models.

CampusContent began its research at this point with the goals of:

- Reshaping the reuse and adaptability of digital learning objects to different application contexts,
- implementing and evaluating reference materials that illustrate design-for-reuse principles for learning objects,
- enabling experienced teachers and instructional design experts to represent best practices in teaching and learning and communicate these to practitioners, and

[1] `http://www.campuscontent.de/`

[2] DFG, the German Research Foundation, provided financial support under code number 44200719.

- supporting the work of course designers, teachers, and students through a coherent infrastructure that enables content sharing across heterogeneous learning management systems.

Later in project, we learned that the inclusion of social networking and collaboration functions could help users to organize communities of practice autonomously, furnish them with collective knowledge spaces and use functions for expressing recommendations, annotations and evaluations.

Typical use scenarios for the project's vision include:

- Author A uses resources from author B and author C, modifies them if licensing conditions permit, and adds her own content or didactic concepts to a seamless composition;
- author B and author C use the same material but for different instructional purposes or in different learning settings;
- a group of like-minded professors establishes a social network, e.g., on the topic "Service-Oriented Computing" and sets up a peer review system for learning materials on this topic;
- didactic experts represent online and blended learning models and didactic scenarios as learning paths or learning designs and publish them.

B. Resolving the ROI Paradox

Learning materials that can be used in different application contexts must be target-group and context neutral. However, good learning content should also be didactically tailored to the actual learning situation and learner group. Baumgartner named this conflict of goals the ROI (Reusability of Objects and Instruction) paradox [3].

We propose to mitigate the inherent contradiction between context-neutral content and the necessity of tailoring learning objects to the needs of the learner by a heuristic principle (see also [5]). This principle is known from software engineering as *late composition*. Adapted to e-learning, it suggests keeping information and didactic context separate at design time and connecting both facets of learning objects only at reuse time.

Fig. 1. Facets of a learning object

The project's model of a learning object was first published in [4]. A *learning object* combines an information object with a didactic scenario and a specific learning objective (see Fig. 1). An *information object* consists of illustrations, pieces of text, simulations, animations, video or audio clips, photos, maps, quizzes, reference works etc. that describe facts, concepts, procedures, processes or principles of a knowledge domain. A *didactic scenario* specifies roles and recommended learning or assessment activities, including learner-learner, learner-tutor, and learner-computer interactions. A *learning objective* specifies the skill development or knowledge acquisition anticipated as the result of a learning process. It connects the information object with the actual didactic scenario.

The components of a learning object are maintained sustainably as relational structures in the repository network the project has built. They can be retrieved and will inspire new combinations and adaptations in community processes, as we hope.

Parameterization is a mechanism also adapted from software engineering. We distinguish two forms: pedagogic and thematic parameterization. *Pedagogic parameterization* aims to equip information objects with parameters that allow its adaption to specific didactic needs. Besides other means, didactic parameterization can be used to realize late composition. We propose a scalar classification of learning objectives relying on Anderson and Krathwohl's taxonomy of cognitive processes [2] (see Section III). *Thematic parameterization* refers to the idea that certain interactive learning objects can be adapted to different topic areas by configuring a set of parameters.

In this article, we illustrate the implementation of these design principles and mechanisms for three types of learning objects that proved to be useful in higher education. We evolved these resources into generic objects from which custom-designed objects can be generated through combination, parameter configuration, and adaptation. Our first example, which is implemented in Java, serves to demonstrate the multitude of combinations we can achieve through late composition and didactic parameterization. Two further examples of generic objects are implemented in Adobe's Flash format. The first one, concept classification, serves to illustrate thematic parameterization, while the second Flash example illustrates the *separation-of-concerns* principle by which different features of an object like graphics design, interaction control, and functionality are treated separately.

C. Portal Edu-Sharing

Besides the conceptual and methodological results presented in this article, CampusContent developed a comprehensive portal, *Edu-Sharing*, that enables the sharing and reuse of digital learning content across heterogenous learning management systems. Versioning of content is supported.

Figure 2 depicts the core components and tools of Edu-Sharing. They can be grouped into authoring and learning support. The heart of the portal is a repository, in fact, a network of repositories because individual institutions may want to operate their own instance of an Edu-Sharing repository. Different instances of the portal repository can be connected through web services to form a distributed network providing a single system view from each participating site.

The distributed repository serves to organize and maintain personal workspaces of registered users and the outcome of authoring activities or content that is uploaded from the user's hard disk. Open interfaces allow different portal operators to connect their preferred authoring tools and learning management systems, while the repository component is standard to ensure interoperability in the network. A range of special editors and two open source learning management systems (LMSes), Moodle [27] and metacoon [24], are included in the standard distribution of Edu-Sharing. Plans and agreements with platform developers exist to interface further LMSes with Edu-Sharing.

The editors serve to produce or compose:

- Different types of data, such as text, video or graphic files representing basic building blocks of information and learning objects,
- assessment questions and tests conforming to the Question and Test Interoperability (QTI 2.0) standard [16],
- learning paths and didactic scenarios,
- learning objectives,
- learning objects, and
- course modules.

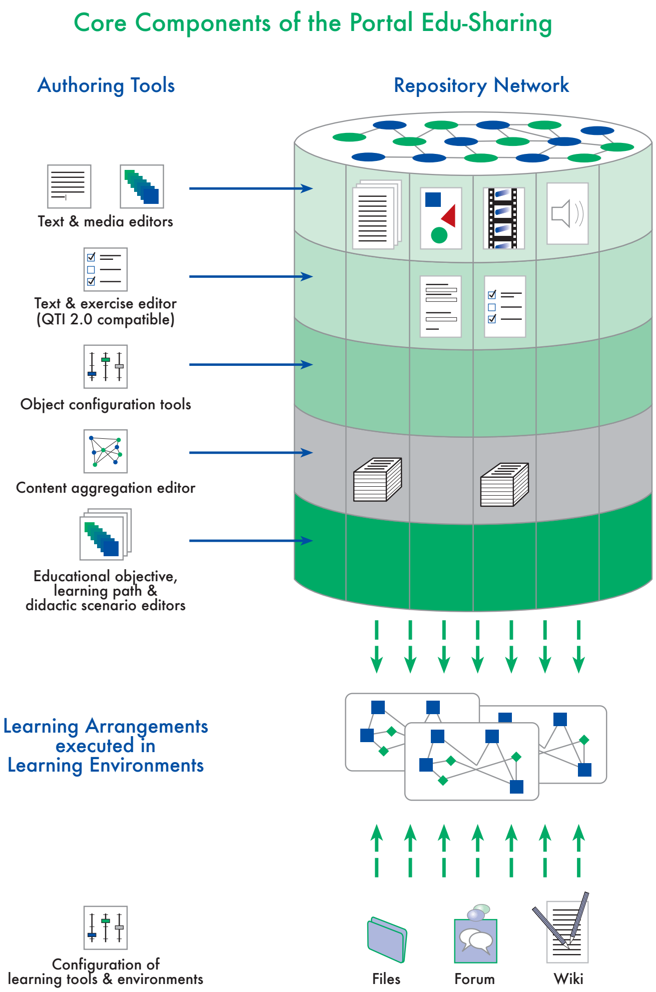


Fig. 2. Components of the portal Edu-Sharing

The repository network also supports learning processes performed in learning management systems or virtual learning environments directly because learning objects used in learning paths or pedagogical scenarios are referenced and executed from within the repository, as a rule. Learning objects can also be downloaded to execute a local copy. However, then the teacher loses the option to request usage data for his or her private instance from the portal's data analysis component (not shown in Fig. 2). Besides the possibility to reference learning objects from the repository network, the integration of repository and LMS offers further options on the LMS side: Search content in the repository; link or insert a learning path, didactic scenario, information or learning object found into the course under construction; store content built in the LMS sustainably in the repository. A license management component, which is also not shown, supports content owners upon upload to associate an appropriate use license with their works in the repository network. The personal workspace of each registered user serves to organize and connect clusters of documents and, more importantly, to share these with others, independent of usage rights and licenses. Thus, the workspaces provide a collaborative environment for communities of practice whose members have similar profiles or build on special trust relationships.

The portal was particularly designed to encourage the sharing and reuse of open educational resources [28]. It builds on the open source content management system Alfresco [1] and the open source portal software Liferay [23]. Currently Edu-Sharing undergoes a pilot use phase with different kinds of user groups at universities, schools and vocational training institutions [22]. By the end of 2009, the software packages developed in the project will be published as open source software to the public at large.

D. Structure of the Article

The article is an invited extension of a paper that was accepted for the International Conference on Mobile, Hybrid, and On-line Learning 2009 [13].

In the following section, we first report on related work. In Section III, we briefly review a well-known educational taxonomy dealing with cognitive aspects of learning. Then we show for a widely used class of models of computation, finite automata, how content can be flexibly combined with learning tasks addressing different levels of cognitive challenges. Section IV explains the components and architecture of the technology supporting our methodology for Flash-enabled objects. This section presents two examples of generic objects. In Section V we sketch an extension of our architecture that aims at raising the degree of adaptability of generic objects through a software component approach. Section VI presents some thoughts about design-for-reuse principles. Section VII explains how prerequisite requirements, learning and assessment activities, learning content, and completion requirements can be combined to learning paths and study courses. Section VIII finally touches upon IPR-related legal issues and discusses how they are addressed in the portal Edu-Sharing. We conclude with a brief summary and an outlook on future work.

II. Related Work

The reuse of digital learning material has been a continuing issue. First, there were a number of initiatives promoting the reuse of educational software. However, their success in practice was limited. The most substantial problems were incompatibilities in language, culture, curriculum, computer-use practices, and didactic approaches of the potential learners and their instructors [9].

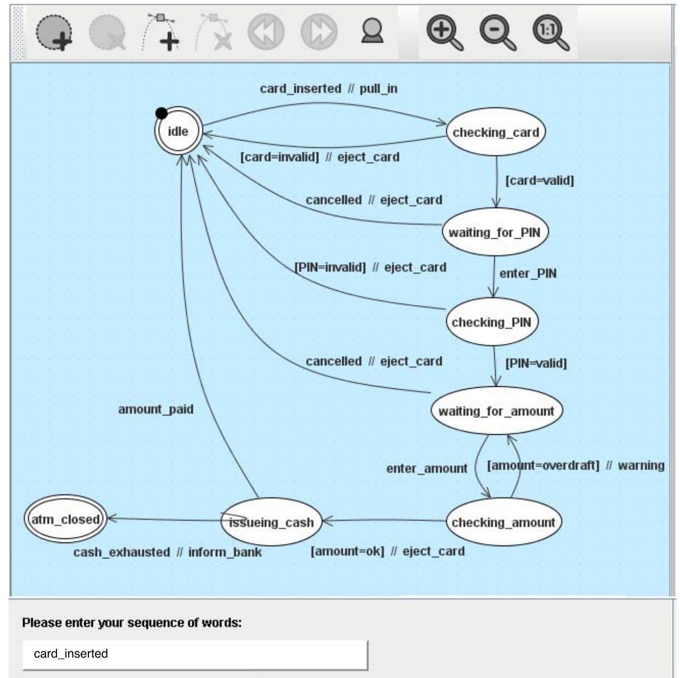
Although David Wiley compared the idea of building educational content from smaller building blocks with object-oriented programming [31], there is no generally agreed development and reuse concept as it exists, for instance, in software engineering. [21] argues that design principles such as encapsulation, cohesion, and decoupling, which allow software developers to develop and maintain objects independently of each other, should be carried over to learning objects to achieve similar benefits.

Boyle was the first who attempted to transfer certain software engineering principles like cohesion and decoupling to learning objects to encourage the production of reusable learning objects [7]. Cohesion among different components of a compound learning object in Boyle's approach is achieved by the fact that all components are focused on a single learning objective. IOs and learning activities with dynamic objects are combined to didactically purposeful learning objects. However, this technology only support white box reuse because a re-user who wants to change a compound object, has to manipulate it with a specific editing tool. More recently, in [18], Jones and Boyle adapted the design pattern approach [11] to learning objects. But this work is less concrete than what we propose in this article. A separation into content and didactic context in the sense of [14] to enhance a learning resource's reuse potential has not been practiced much.

III. Didactic Parameterization of Educational Resources

David Wiley seems to be the first who discussed the connection between learning objects and instructional design [31]. To achieve a practicable solution that seamlessly integrates the information and instructional facets of learning objects, we studied various educational taxonomies including Benjamin Bloom's well-known taxonomy of educational objectives [6] and Anderson and Krathwohl's more recent revision of Bloom's taxonomy, AKT for short. AKT aims to accommodate new insights in cognitive psychology, curriculum and instructional design, and assessment. Both taxonomies describe six levels of cognitive performance with increasing complexity. In AKT, they are labeled: "Remember", "understand", "apply", "analyze", "evaluate", and "create". "Remembering" requires students to recognize relevant knowledge or recall it from long-term memory, while being able to "create" refers to the ability to devise a plan, put building blocks together to form a coherent or functional whole, to reorganize components into a new structure, or produce new artifacts.

In this section, we use this taxonomy to qualify and relate learning tasks and activities and combine them with different instances of information. We call this *didactic parameterization* of information objects and illustrate its use for the topic area "finite automata". A finite automaton (or state machine) represents an abstract mathematical model of a physical of mental machine with a memory. Finite automata are frequently used as modeling tools in different disciplines, including computer science, engineering, linguistics, or biology. Even learning designs have been modelled with finite automata. A finite automaton can be represented as a mathematical structure, a visual state transition diagram (see Fig. 3), or a transition table. In addition, a finite automaton is a computing device that accepts a regular language.



Task: Find a sequence of input strings leading to an accept state.

Fig. 3. Learning object with state transition diagram as information object

Figure 3 shows a learning object including a state transition diagram and a learning task. In AKT, this learning object would range at the second lowest cognitive process level "understand". To this end, we assume that a student has studied the basics of finite automata and is about to test his or her learning achievements. What we expect from students to recall here is simply the concept of finite automata, their behavior in terms of inputs and state transitions and their relationship to regular languages. Students can enter their solution in the window at the bottom and their input is immediately checked based on standard algorithms.

Following the late composition principle, the learning object depicted in Fig. 3 is maintained as a relation rather than a closed object in the repository network underlying the portal Edu-Sharing. The relation consists of an information object (here: the representation of an automaton in the form of a state transition diagram) and the specific didactic context (here: the learning task description).

The flexibility of this approach derives from the fact that it allows us to combine a single information object with different didactic contexts that are organized along Anderson and Krathwohl's or any other suitable educational taxonomy. For instance, the automaton in Fig. 3 could have been used by other teachers in combination with the following learning tasks that address different cognitive levels:

1) **Remember**: Define the mathematical structure of the automaton shown in Fig. 3.
   Determine whether the diagram denotes a graph, a tree, a Petri net, a communication protocol, or a finite automaton.
2) **Understand**: Provide a sequence of strings that leads the automaton in Fig. 3 to an accept state.
   Determine whether the automaton in Fig. 3 will accept the following sequence of strings: card inserted, [card=valid], enter PIN, [PIN=valid], cancelled.
   Develop a transition table that is equivalent to the automaton in Fig. 3.
3) **Apply**: Provide the regular language that is accepted as input by the automaton in Fig. 3.
4) **Analyze**: Assume that the automaton in Fig. 3 models the behavior of an automatic teller machine. Determine how many states and transitions need to be added to the automaton in Fig. 4 to model the case that a bankcard is withdrawn after three failed attempts to enter a PIN. Expand the model correspondingly.

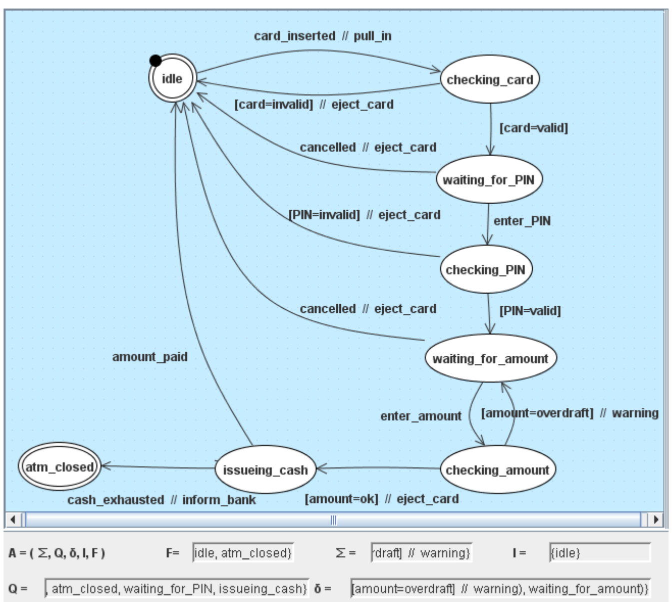


Fig. 4. Student view of the visual ATM model; the student has defined mathematical structure corresponding to the graph

A user who finds, e.g., the learning object shown in Fig. 4, will also be informed that this information object has been combined with other didactic contexts, which are listed above.

Conversely, these learning tasks could have been used together with another automaton that better fits into the larger context of another teacher's course. In Fig. 5, for instance, the learning task from our first example in Fig. 3 has been used in combination with a different information object. It models a simple bottle sorting machine for large and small bottles (lb, sb) that are, e.g., delivered via a conveyer belt and need to be sorted by removing bottles one by one from the belt and dropping them into a box for small or a box for large bottles, respectively (rsb, rlb).

Once an object like this is found, all combinations of this object in other contexts are listed to stimulate authors and re-users to provide parameterized objects and build on others' work.

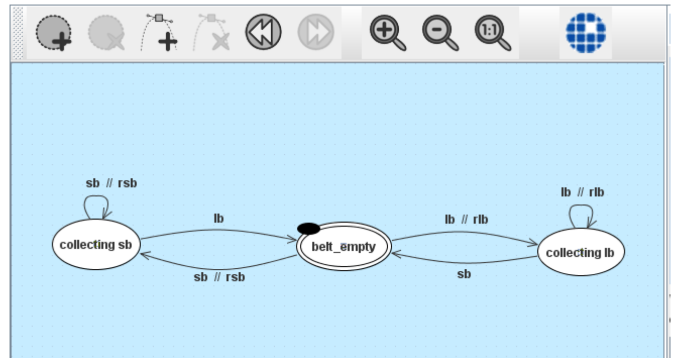


Fig. 5. Model of a bottle sorting machine reusing the didactic scenario from Fig. 3

The portal Edu-Sharing includes a Java-based editing tool that allows users to modify automata and create new ones (as state transition diagrams, transition tables or mathematical structures), to adapt an existing didactic context or define a new one, and recombine existing or new elements. Students can also execute a specified input sequence on a given automaton to determine whether their understanding of an automaton's behavior is correct. Figure 6 illustrates the preview an author can select to test the student's perspective before publishing his or her work. For automata-based learning objects associated with lower level cognitive tasks, the students' response can mostly be checked automatically based on the semantic equivalences between different representations of automata.

Of course, modifications to information objects and didactic context are only enabled if the re-user owns the right to do so (see also Section VIII).

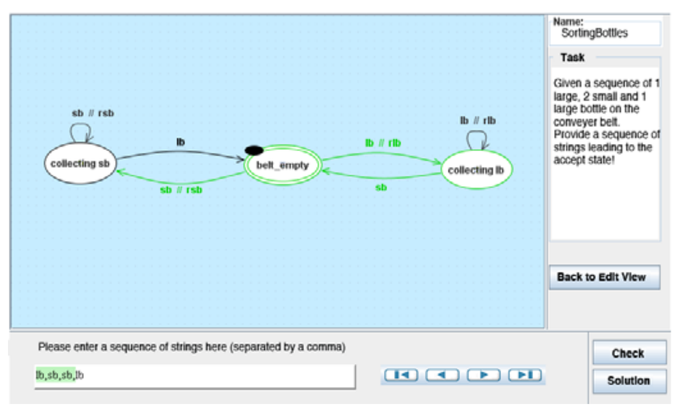


Fig. 6. Executing the model of a bottle-sorting machine in preview mode

We plan to build similar environments for graphs and, as special types of graph, trees and Petri nets. These objects share important properties with finite automata that support a didactic parameterization: They have a visual representation that can be used to model a rich set of real-world problems ranging from social network analysis over routing problems in street or communication networks to coloring problems. This bears the potential for defining higher-level cognitive tasks of type application, analysis, evaluation, and creation. They come with a well-defined mathematical theory and are related to other theories like linear algebra. Finally, the theoretical underpinning provides the basis for a rich set of traversal and manipulation algorithms, which can be exploited to test a student's response automatically.

IV. Design and Implementation of Adaptable Interactive Learning Objects

Adobe's Flash is a popular media type for implementing animated learning objects. Advantages include powerful animation and interaction capabilities, availability of Flash players and plug-ins on many operating systems, and ease of distribution and integration into interactive courseware. However, apart from didactic competence, the development of high-quality Flash animations requires know-how in media design and technical animation skills. Therefore, it will often be a better choice for a course author to reuse and – if necessary – adapt suitable animations from others rather than developing them from scratch.

In this section, we will illustrate the design of customizable Flash animations with two examples of learning objects that previously proved to be useful in higher education. The advantages and drawbacks of the two methods employed will be discussed in detail. We also describe the architecture of the technology used, which relies on Adobe's Flex framework.

A. Adobe Flex Framework

Flex is a new technology proposed by Adobe. It aims at providing a free, open source framework for building highly interactive Web applications. Flex applications are compiled into Flash (.swf) files that can be deployed and run consistently under major browsers and operating systems. The Flex framework provides a standards-based language and a programming model that supports common program components, in which user interface (UI) design and client logic implementation are clearly separated. MXML, a declarative XML-based language, is used to describe UI layout and behaviors. ActionScript 3, a powerful object-oriented programming language, is used to create client logic. These features of the Flex framework provide several possibilities to develop reusable animated and interactive learning objects.

In Flex, Flash animations can be generated by compiling the MXML text file, which may represent the template for a family of animated objects. If properly parameterized, each template can be configured differently by different instructors to accommodate their individual didactic context. The configured template can then be compiled into different versions of the generic Flash animation. As the Flex framework takes a component-based programming paradigm, a Flash animation itself can become a programmable object. This provides the basis for developing information objects that are largely free from context and expose possible animations through a programmable interface. The re-user then only needs to take care of the desired didactically meaningful interaction behavior. Figure 7 depicts the conceptual architecture of the proposed method. Concrete application examples are presented in the following two subsections.

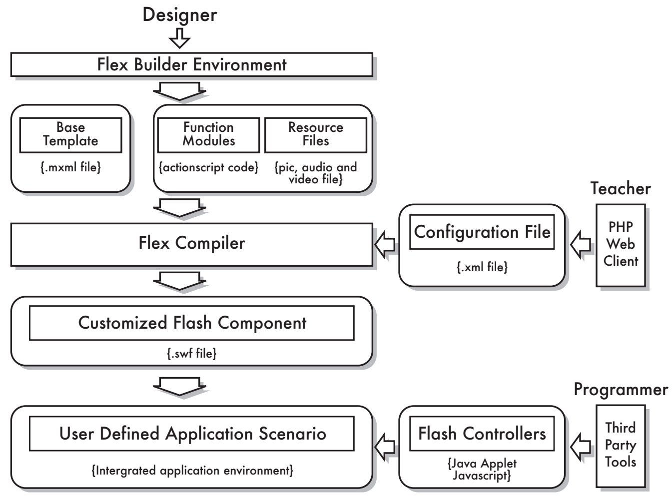


Fig. 7. Conceptual architecture of Flex-based learning object design and customization

B. From Object to Template

Besides the advent of Flex, a motivational element behind our approach was the desire to reuse a simple interactive Flash animation in a different thematic context without the need for editing the Flash file. A simple example is shown in Fig. 8. This animation aims to test the following educational objective, which would reside on the comprehension level of Bloom's or Anderson and Krathwohl's cognitive taxonomies:

> Given a set of concepts that were raised prior in this course in a case study illustrated by a number of authentic car rental scenarios, the student in a beginner course on object-oriented programming will be able to accurately sort 20 concepts into the three categories 'object', 'attribute' and class' within less than two minutes.

Figure 8 depicts the students' view of an interactive Flash animation currently in action. When the animation is started, a list of the concept will begin to move down the screen from top to bottom and thereby increase in size. The student has the task to pick the terms one-by-one with the mouse cursor and drop them into one of the three folders. This activity continues until all concepts have been sorted properly or the student gives up. Concepts that were dropped in the wrong folder will reappear in the scroll-down list.

As conceptual knowledge is important in any scientific and technical field, we designed a configuration environment for building concept classification objects from a Flex template. In this redesign process, we also included further parameters to control the interaction such as a timer, an error counter and a scrolling speed parameter.

Fig. 8. Screenshot of a Flash animation for concept classification

C. Customizing Parameterized Animated Learning Objects

Obviously, the concept classification animation can be used in a range of subject areas and disciplines including biology, software engineering or physics. To reuse and adapt the original Flash animation, it is, however, necessary to have access to the source file, the right to modify it, a Flash authoring tool or IDE, and sufficient Flash skills to implement the desired changes. This would correspond to white-box reuse in software engineering, which is the core of open source developments.

In this section, we will illustrate how the generalization and customization of such a learning object can be achieved through thematic parameterization. First, we need to generalize the educational objective to make it independent from the concrete case:

> Given a set of sample concepts and definitions of subject-related concept categories, the student will be able to accurately sort these concept into a predefined number of categories in a predefined time or with no more than m false classifications.

Further we need to create a template that allows the teacher to name the n concept categories desired, n sets of concepts to be used as test cases, one for each category, and n icons visualizing these categories. To provide additional flexibility, we introduce a range of parameters for defining

- The number of errors allowed,
- the maximum amount of test time,
- the rolling speed,
- the explanatory text including hints how to use the animation,
- the educational objective,
- background color, font, minimal, maximal text sizes, and other visual attributes.

To indicate the number of errors made and the time used for the test, we also need an error counter for each category and a timer.

Figure 9 partly shows a configurable Flex template implementing these features. For pragmatic reasons like screen presentation and complexity of use, we allow between two and six different categories. The preferred icons representing concept categories can be uploaded from the teacher's computer and textual elements can be copied or typed into the text windows named "Concepts in Category i". Once all desired modifications are made, the re-user can activate the "preview" button to view the customized animation. The configuration data will be written into an .mxml file from which the server-based Flex builder will compile the new Flash animation, which is presented at the client side. The final Flash version can be downloaded or – in the case of Edu-Sharing – be stored in the portal's repository and referred to in different courses.

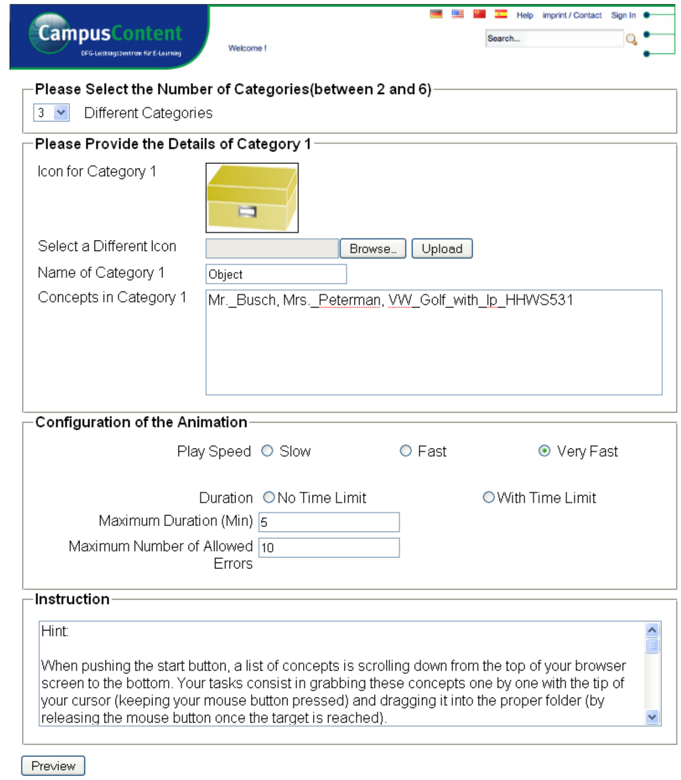


Fig. 9. Configuration interface of the parameterized animation

Figure 10 depicts a customized version of this template that is used in our course "Object-oriented Programming" in place of the original version shown in Fig. 8.

What Edu-Sharing users will find when searching the repository, are just fit-for-purpose objects like the one shown in Fig. 10. Compared to many other objects in the repository, the ones derived from a template carry a button "Customize" at the bottom, which suggests that such objects can be adapted. When clicking this button, the Flex-based template editor will be launched and the re-user can manipulate its parameters and produce animations that satisfy their needs.

D. Reuse of Animated Learning Objects as Software Components

The parameterization method discussed in the previous section provides a simple and effective way to customize animated learning objects without requiring special capabilities from the re-user. But it also exhibits limited flexibility because

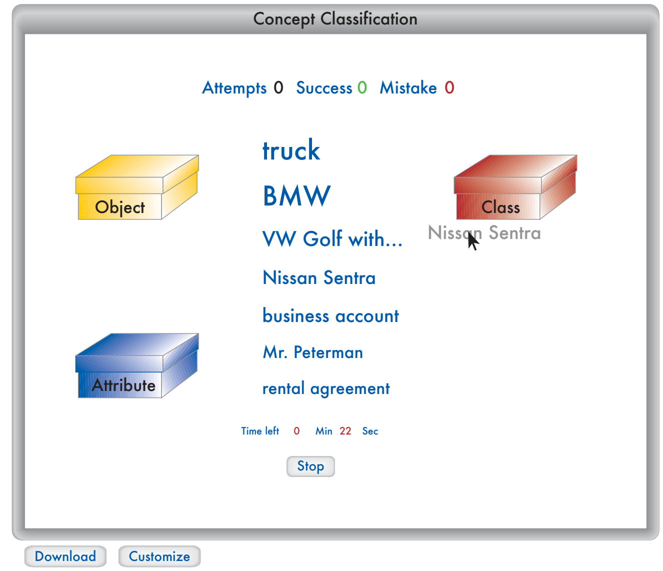


Fig. 10. Screenshot of configurable animation ”Classify domain concepts”

the re-user has almost no possibility to change the didactic design. This is due to the fact that all the possible interactions between the flash animation and the user are hard-coded in the .mxml file of the base template. The teacher configuring it can change the appearance and set certain parameters limiting the animation behavior but he or she cannot modify the application logic.

To overcome this restriction, we propose a second reuse method. As already mentioned, the lowest layer of Fig. 7 suggests that the Flex framework allows each compiled flash animation to be used as a software component that may interoperate with other components. To further enhance an animation's adaptation capabilities, we propose to just define generic animation movements for the base template rather than a particular interaction sequence. In addition, a set of functions to activate such movements is exposed to the environment of use in the form of application programming interfaces (APIs). Re-users can use these APIs to define their preferred control sequences accommodating different didactic scenarios without the need to touch the Flash template or the resulting animation.

Fig. 11. Customizable ”Bottle” Flash Animation

In Fig. 11, we show a Flash animation for a variable set of bottles. A simplified API for this animation is listed in Table I. Animation and API can, for instance, be used to visualize the behavior of the bottle sorting machine discussed in Section III and Figs. 5 and 6. We could write an algorithm that creates

TABLE I
LIST OF THE KEY APIS

Operation	Intended meaning
setBottle(s,c,i)	Create new bottle of size s and color c and place it at position i
getBottleNumber()	Get number of bottles
isEmpty(i)	Test whether position i is empty
getSize(i)	Get size of bottle in at position i
moveBottle(i,j)	Move bottle at position i to position j
highlight(i)	Highlight color of bottle at pos. i
removeBottle(i)	Remove bottle at position i

large and small bottles one by one and moves them from left to right from position 0 to 9 and then removes them again one by one. This behavior would simulate a conveyer belt. Then we could define that a small bottle has to be removed from position 6 and a big bottle from position 8 to simulate their sorting into different boxes. The transitions in Fig. 5 that are labelled with the input strings sb and lb (for small and large bottle, resp.) and have no output string could then be equated with a ”detect bottle size and move right by one position” operation for all bottles left of position 6 or 8, respectively. The transitions labelled sb//rsb could be equated with operation removeBottle(6) and those labelled lb//rlb with operation removeBottle(8).

In another context, we could use the ”Bottle” animation in combination with a sorting algorithm controller that allows us to apply different sorting algorithms to an unordered collection of bottles of different size. Students could be asked to observe a sorting animation and determine the actual algorithm that was applied and reason about their insights.

To give another example of the advantages of decoupling visual representation and animation control, Fig. 12 shows a combination of a map of Germany and a controller implementing different graph traversal algorithms including breadth-first, depth-first, and Dijkstra's algorithm. The map shows connections between major cities, which represent the nodes of the graph, while connections are visualized as edges.

A learning task could then be to determine the shortest route between two cities A and B, where the distance is determined be the number of edges between A and B. Alternatively, the edges could be labelled with kilometers or another metric and the task would be to determine the cheapest connection between A and B. The screenshot of the map shows a situation in which Dijkstra's algorithm is used to measure the distance between Hamburg and Munich.

Each connection between two cities can be highlighted throughout the animation using the APIs of the animation. Through these APIs, a third-party program can also query the weight or distance associated with each connection and the currently selected node(s).

The lower part of Fig. 12 illustrates how a third-party program can make use of this animation. This behavior can be used to

- visualize an algorithm's behavior in the form of changes on the map,
- let a student control the manual execution of the al-

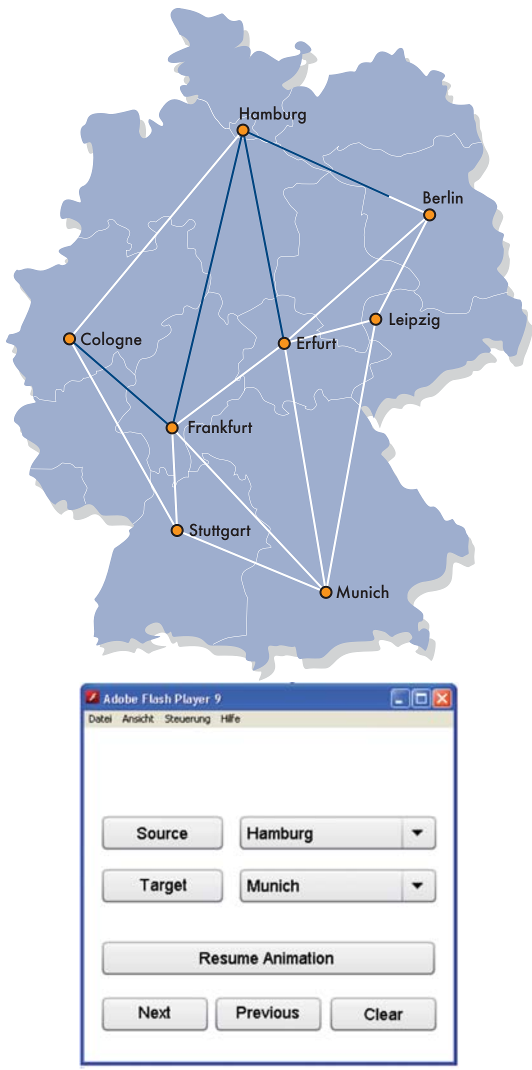


Fig. 12. Interacting with different graph algorithms

gorithm by clicking on selected edges in the proper sequence, or

- implement other learning tasks.

A third-party program can control the whole animation process through the APIs provided by the graph animation.

V. Animated Learning Objects Viewed as Software Components

In the previous section, we discussed two different approaches for developing reusable interactive learning objects. To enhance their reuse potential, we suggested a further separation of concerns. We proposed to define the visual appearance of an animation and an unconstrained behavior with the help of a Flash editor and implement meaningful behaviors in the form of controllers that are interfaced with each other through APIs. This approach has the desired side effect that both components can be maintained separately as long as the interface remains stable. Again, this is a design principle that has been exploited successfully in component-based software engineering.

As we pointed out in the introduction, learning object development is a complex process that involves different competencies such as instructional design, media design, programming, and domain expertise. It is unlikely that one person owns all these skills. Therefore, we believe that an effective reuse paradigm for learning objects should be leveraged to a higher degree of productivity by using the best fitting technology and flexibly organizing the cooperation of necessary competence holders. Based on his or her own expertise, a re-user can choose the corresponding level of customizing learning objects. To support such processes, we aim to provide a collaborative software environment in which re-users with different expertise can work together seamlessly.

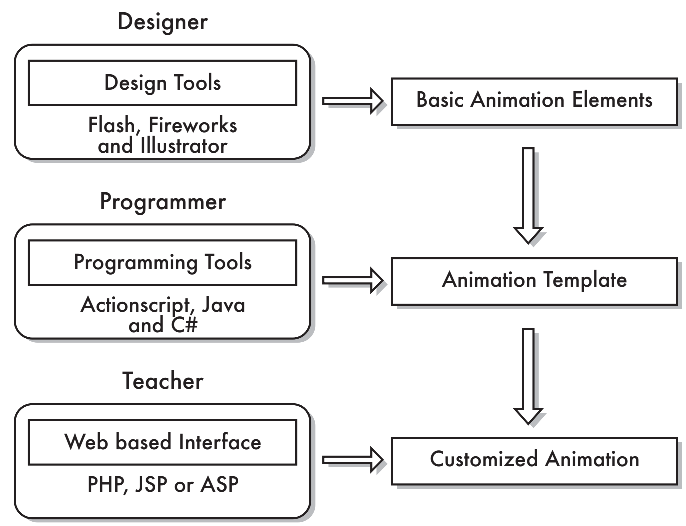


Fig. 13. Reference framework for scalable reuse of animated learning objects

In Fig. 13 we present a reference framework for the scalable reuse of animated learning objects described above. Within this framework, re-users at each level are supposed to work largely independently while at the same time being able to benefit from their mutual contributions. This framework can be realized with the FLEX environment and the possibilities that other tools provide. Java applets have been investigated to some extent as control components for Flex generated animations. In programming education, for example, this feature would allow students to implement their own animation control algorithm in Java.

While the framework sketched above is technically realizable, the tasks of re-users at each level are related to each other and cannot be separated so clearly. For example, to program the application logic, the developer must communicate with the instructor to understand the didactic scenario and requirements to be satisfied. Within a closed group, this may be easier to solve. However, when considering this issue in the context of an open collaborative platform, it will be difficult for re-users with different backgrounds to express their requirements and locate the appropriate resources. We aim to address this problem by defining a unified description schema that can be understood and used by all persons involved in such a collaborative design process.

VI. Design for Reuse and White-box Adaptation

To illustrate the advances that can be achieved through a continuous strive for reuse potential, we want to report on the evolution of modular learning materials for introductory statistics courses for different disciplines.

A closely collaborating colleague began with an initial version of a multimedia course that was built with a proprietary authoring tool and was delivered on a CD. The reuse potential was close to zero, even for the author himself. After spreading the message about reusable content, a new version was produced in HTML, including a large number of Java applets, audio and video files, and animations. In principle, reuse possibilities had increased, but only if the author was willing to provide the source code of his applets and other mulimedia components. The explanation is that none-HTML components in this course are just referenced from locations to which third parties have no access.

To overcome this weakness, the author now offers a collection of outstanding interactive learning objects in statistics [25] that can be downloaded from his home page [26] or referenced from within the Edu-Sharing repository. Many of these objects include textual and spoken explanations in German, English, French and Spanish, which suggest another mechanism to increase reuse potential: language parameterization. Due to the quality of this work, some objects have been translated into Japanese and are used at Japanese universities. This work is closely connected to similar work of other professors teaching statistics at other German universities. Their joint work towards the goal to develop the foundations of a new approach towards statistics education (New Statistics) was financially supported be the German Minister of Research, and the outcomes are currently used and maintained by 10 German universities.

Figure 14 shows a snapshop of an interactive experiment taken from [26]. It refers to the Gini coefficient or index that is a statistical measure to represent unbalanced distributions. It can be used to investigate and explain the important question of equal distribution of income, wealth, power and influence, or markets that are relevant in different disciplines, including business management, economics, or political sciences The situation of the experiment depicted in Fig. 14 indicates a deviation from equal distributed in the shaded area underneath the $45°$ line. This state could be the result of a student who tried to solve the following learning task:

> Interpret the Gini coefficient and demonstrate what it represents by modifying the sliders in the animation such that you obtain a deviation from the ideal Lorenz curve. Then identify the percentage of superstores that generates 50

Although this object is extremely well designed, a second glance reveals some potential for improvement towards a higher degree of reusability. Both paragraphs of the explanatory text (and audio) refer to a concrete example, which might not fit the context of another author's course or lecture so well. Others may want to add further recommended interactions and learning tasks. To enable this, they need to have the right from the author of this experiment to change its content, they need access to the source code and they need a proper tool to operate on this source. We call this *white-box adaptation* as opposed to the *black-box adaptation* we discussed in previous sections. Black-box adaptation only manipulates the interface of an object, while white-box adaptation modifies the object's interior and as such it ressembles the open source software development approach.

Edu-Sharing is open to all types of adaptation and reuse. Only content authors can impose constraints with the type of use license they declare and a lack of proper editing tools or skills on the re-user's side can prohibit white-box adaptation.

In the following section, we discuss a practice-oriented didactic model that allows us to cure the flaws of the Gini experiment by separating out those parts that are likely to be changed by re-users into the different facets of a learning path. However, before doing so, we summarize a few observations aiming at good design of information objects. Some of these principles are specific to the topic; others have been inspired by "design-for-reuse" principles in software engineering.

- Avoid verbal references to external sources;
- avoid hyperlinks to resources not accessible in the given virtual learning environment, here, the portal Edu-Sharing;
- find and isolate topics, concepts and notations of expected variability and try to use parameterization to handle contextual variability;
- constrain an information object to common invariant content;
- use aggregation and hierarchy to compose more complex objects from simpler ones;
- design information objects as if they were stand-alone products;
- capture context, educational aspects and documentation in proper facets of learning paths (see next section).

VII. Learning Paths

In an early phase of the CampusContent project, we studied the IMS Learning Design (LD) standard [15], [20] and a few prototype versions of LD editors intensively with the intent to use this standard as a basis for technology development. The experiments with these editors were, however, not encouraging because they exhibited too many usability weaknesses. However, more importantly, in many conversations with potential Edu-Sharing users, we recognized that the LD standard is not yet popular in educational practice. As LD will be better received, a usable LD editor will be included in the portal Edu-Sharing.

For now, we decided to adopt a pragmatic approach to codify learning processes in the form of learning paths. A *learning path* is a sequence of learning phases an individual follows to acquire knowledge, skills and competences. This approach builds on many of the concepts promoted in LD but presents them in light version. Our phases just distinguish two roles, teacher and learner, while LD allows the definition of arbitrary many roles. A phase involves a learning objective, resources and activities. Activities can be supported by e-learning, cooperation, and communication tools. Each phase

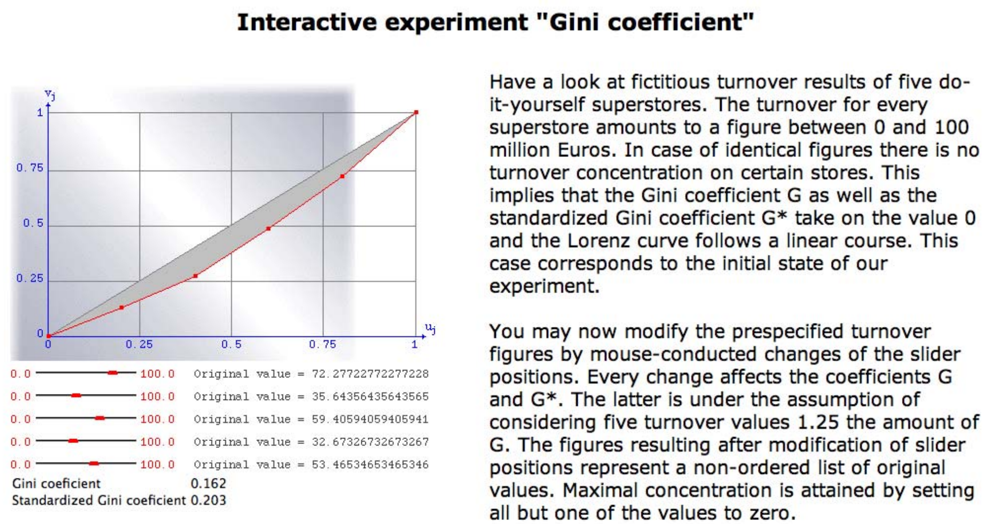


Fig. 14. Interactive experiment to deepen the grasp of the concepts Lorenz curve and Gini coefficient

may include information or learning objects that are thereby aggregated to higher levels of granularity.

Figure 15 shows the first phase of a blended learning approach that is based on the well-known project-based didactic scenario. The complete scenario consists of seven consecutive phases:

1) Team formation and initial setup (the phase depicted in Fig. 15);
2) requirements acquisition and evaluation;
3) draft design;
4) comparison and evaluation of different team solutions, refinement of preferred design solution;
5) implementation and testing;
6) test evaluation;
7) archival of project results.

The first three phases are organized as self-study phases for geographically dispersed students who communicate and interact with each other and with the tutors using the Edu-Sharing's workspaces, email, a wiki, and a forum. Phases 4 and 5 are organized as face-to-face meetings in a location that provides access to professional software engineering tools. Phases 6 and 7 are again self-study phases. Earlier versions of this scenario have been used (with other means) several times by the first author to conclude a two-semester distance-learning course on software engineering.

The icons in the lower right part of the ”Student Activities” pane indicate that the students' activities are supported by a wiki, a forum and a document folder. The teacher who adapted this scenario to her or his needs has specified this. Resources comprise learning objects, learning units, and arbitrary types of documents, while activities include individual and group activities, interaction and communication activities. The different tabs may include links to online material stored in the repository, an Edu-Sharing workspace, or elsewhere on the Web. It can also specify offline resources to look at in this phase.

Reusable scenarios should be independent of a particular

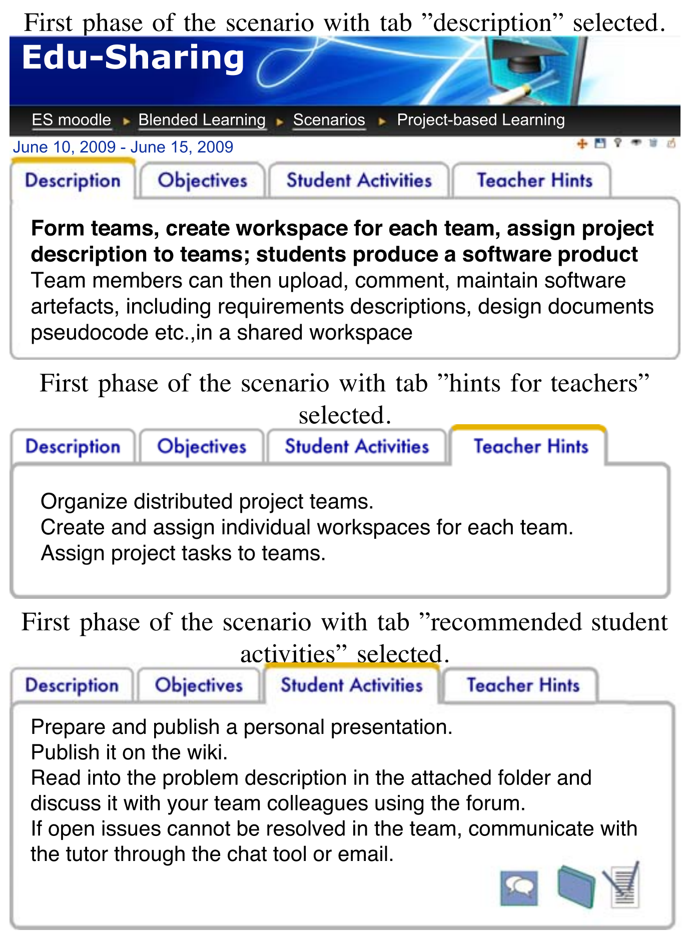


Fig. 15. Different sections of the top part of a blended learning scenario maintained in Edu-Sharing and presented in Moodle

discipline. Edu-Sharing offers a growing number of mature scenarios of different granularity that we adapted from literature and codified using the concepts discussed in the previous paragraph. Fine-grained examples include devils advocate, active structuring, flashlight, brainstorming, concept mapping, think-pair-square and webquest. More complex scenarion that typically rely on tool support include case study, jigsaw classroom, puzzle method, strategic problem solving and others. They are published in the form of generic templates, i.e., without specific resources and tools, in the portal. To facilitate search and finding, these templates are supplemented with appropriate metadata, which have been defined by the project CampusContent (LOM and Dublin Core are metadata standards offered to decorate information objects).

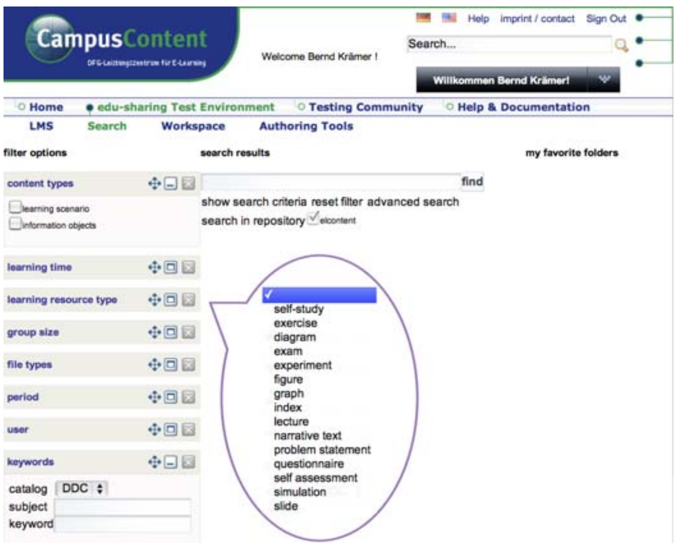


Fig. 16. Searching for content in the repository

Figure reffig:search shows a few filter options that Edu-Sharing users can select in the process of searching information, learning objects or scenarios. The window that pops up when selecting the filter ”learning resource type” is shown in the ballon in the center pane. Re-users who find such scenarios and want to adapt them to their needs, can edit them with the help of Edu-Sharing's scenario editor. We expect that such scenarios may inspire educators, who had no clear idea before of what a didactic scenario is, to try them out in their own teaching.

VIII. Open Content, Intellectual Property Rights and Licensing

An open environment for exchanging intellectual property requires clear and legally well-defined regulations to ensure that the interests of both rights owners, i.e., the authors of information and learning objects and of didactic scenarios, and re-users like trainers, teachers, lectures etc. are respected and served. Content authors want to maintain their right to decide what others are allowed to do with their intellectual property. Potential re-users need the certainty of the law and more than just trust in the availability of third-party content. It is also in the interest of portal operators to limit liability to their sphere of influence.

”Open content” initiatives and their specialization ”open educational resources” have been inspired by the open source software (OSS) movement that promotes licenses allowing the free access to source code and its unlimited non-commercial distribution, reuse and adaptation. In the late 1990s Wiley and others adapted this definition to digital content of various types including text, image, graphics, audio, video, animation and the like. In contrast to the ”all rights reserved” claim of the classical copyright, open content requires subtly differentiated possibilities for organizing use and modification rights. This is addressed by a range of dedicated license models including the GNU Free Documentation License [12], originally designed for documenting OSS, the OpenContent License [30] or the more recent and relatively popular Creative Commons License [10].

By default, the portal Edu-Sharing supports Creative Commons but other license models can be made available as needs of certain user groups arise. A license manager pops up whenever new content is uploaded in the portal's repository network. It allows content owners to select a proper license and informs content users about the license conditions of particular content objects found. In addition, when composing several information or learning objects to a larger units of learning, the license manager detects incompatible licenses imposed on compound objects.

IX. Conclusion and Future Work

Already in his early seminal paper from May 1975 entitled *Guidelines for a general didactic concept for the development of study materials in distance education*[3], Otto Peters [29] stressed the need for adapting didactic elements like objective, topic, method, and media to learning situations found in distance learning. This leads to the intertwining of learning objectives with instructional methods and media, whereby the media have to be tailored to the actual setting. The main functions of technical media include content representation, contact medium, and illustration and visualization material, which should be systematically evolved in media didactics that is concerned with the planned, targeted and reflected use of technical media for educational objectives and purposes. The concept of learning objects promoted by CampusContent conforms to Peters' idea of intertwined facets consisting of information, a learning objective, and a didactic scenario that describes what the object can be used for and how learners will interact with it. To provide a high degree of flexibility, we allow re-users to dissect a learning object and recombine it differently.

This article particularly focused on methods and a scalable framework for developing and personalizing customizable interactive learning objects. Three case studies based on real applications have been presented. The technologies used include Java and the Adobe Flex environment.

[3]Title translated by the authors. The original paper is written in German.

The core motivation behind this work is a contribution to the realization of a knowledge building, sharing and improvement cycle (see Fig. 17), which was inspired by [8] and in which:

- educational content authors are supported effectively in the process of creating added-value adaptable information objects and representation of best didactic practices in the form of scenario templates that can be flexibly associated with information objects and learning objectives;
- lecturers and teachers are encouraged to review and analyze the educational knowledge captured in learning objects and thus learn from the knowledge of their peers;
- lecturers and teachers are enabled to adapt and integrate the knowledge of peers and integrate it in their own knowledge.

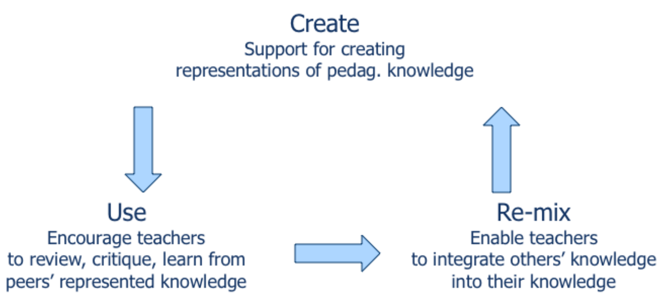


Fig. 17. Educational knowledge building and sharing cycle

In addition, we presented a technical infrastructure and portal, Edu-Sharing, that supports the sharing and reuse of learning objects and mature didactic scenarios. The infrastructure includes a network of repositories for sustainable storage and effective finding of reusable content, a range of tools for didactic scenario and content authoring, a licence manager, and community support including personal workspaces that can be shared with trusted peers. The portal can be easily interfaced with existing learning management systems (LMSes). The open source LMSes Moodle and metacoon will be included in the standard distribution. Others like Olat[4] or Ilias[5] will be supported soon. Open interfaces also allow the adaption of external user management systems and external data stores maintained by commercial providers, such as schoolbook publishers. The license manager will control accesses to such external stores based on access rights defined in local or foreign user management systems. For instance, a school may maintain information about user rights to access learning materials from Klett International. This information is used transparently by the license management component of Edu-Sharing to route an access to object O of a student from that school to Klett's database if the student's teacher owned and passed the right to access O to her students.

Currently, the portal Edu-Sharing undergoes a pilot test with different user groups including university lecturers, high-school teachers, and vocational school teachers to evaluate different use scenarios and identify bugs and usability flaws in the software. A revised version of the portal software will go public by the end of 2009. Therefore, we are currently lacking sufficiently large user groups to provide more mature evaluation results.

The anticipated added value of the project resides in the benefit lecturers and course authors gain from relying on previous work of their colleagues in subject areas bearing similarities in content and instructional design. As a result, users will have at their disposal an additional capacity for the improvement in specific areas of teaching. However, to achieve this goal, a critical mass of content and active participants in every subject is necessary. Therefore, networking of institutions and community building are currently major tasks of the CampusContent project management, besides supervising the pilot application phase.

Critics may argue that there is not much rich content available in (higher) education. This is even true in view of the open educational resources movement, which evolves into a world-wide community effort that includes milestones such as MIT's Open Courseware initiative or, more recently, the Open University's OpenLearn initiative and others. In addition, the coarse granularity of these resources limits their potential for reuse in other contexts. But this is presumably just a contemporary observation. In an interview with Richard Katz Andy Lane, one of the key figures behind OpenLearn, stated [19]: "...we shift from delivering relatively static content embedded in books and printed materials to delivering dynamic content via the Internet", and a bit further down the lane he said: "We are investing in more multimedia content, more simulations, more animations and video ...". Therefore, there is hope that the situation will improve as the symptoms are obvious.

[4]https://www.olat.uzh.ch/
[5]http://www.ilias.de/

REFERENCES

[1] Alfresco: Open Source Enterprise Content Management System. Retrieved September 27, 2009, from http://www.alfresco.com/

[2] L. W. Anderson and D. R. Krathwohl. *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*. Allyn & Bacon 2000

[3] P. Baumgartner. ROI: Reusable Object and Instruction Paradox. Keynote at the 9th European Conference on Media in Higher Education (GMW), Graz, Austria, 2004

[4] P. Baumgartner and S. Heyer. The Trinity: Towards a Model of Learning Objects. Draft paper, 2009, retrieved September 27, 2009, from http://www.google.de/url?sa=t&source=web&ct=res&cd=8&url=http%3A%2F%2Fwww.peter.baumgartner.name%2Farticle-de%2Ftrinity_model_of_lo.pdf%2F&ei=bzG2Sq_DNdqI_AbEmpWoDQ&rct=j&q=baumgartner%2C+trinity&usg=AFQjCNHgIJt6RBgxuzqrSn4vz5NHLZAQ-g

[5] P. Baumgartner and M. Kalz. Wiederverwendung von Lernobjekten aus didaktischer Sicht. In *Auf zu neuen Ufern! E-Learning heute und morgen*, 97-107, Medien in der Wissenschaft, Vol. 34, Waxmann, 2005 (in German)

[6] B.S. Bloom. *Taxonomy of educational objectives, Handbook 1: Cognitive domain*. New York: Longmans Green 1956

[7] T. Boyle. Design Principles for Authoring Dynamic, Reusable Learning Objects. *Australian Journal of Educational Technology*, 19(1):46-58, 2003

[8] J.S. Brown and R.P. Adler. Minds on Fire? Open Education, *EDUCAUSE review*, 17-32, Jan/Feb 2008

[9] B. Collis and A. Strijker. Technology and human issues in reusing learning, *Journal of Interactive Media in Education*, vol. 4, pp. 1-32, 2004

[10] Home page. Retrieved September 27, 2009, from http://creativecommons.org/

[11] E. Gamma, R. Helm, R. E. Johnson, and J. M. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1994

[12] Free Software Foundation (Ed.), *GNU Free Documentation License*. Retrieved September 27, 2009, from http://www.gnu.org/copyleft/fdl.html

[13] P. Han and B.J. Krämer. Generating Interactive Learning Objects from Configurable Samples. In: Proceedings of the International Conference on Mobile, Hybrid, and On-line Learning (eL&mL), IEEE Computer Society, 2009.

[14] S. Heyer. Pedagogical enrichment of information objects. *Journal of Integrated Design and Process Science*, 10(3):33-45, 2006.

[15] IMS Global Learning Consortium. Learning Design Specification. Retrieved September 29, 2009, from http://www.imsglobal.org/learningdesign/

[16] IMS Global Learning Consortium. Question and Test Interoperability Overview. Retrieved September 27, 2009, from http://www.imsglobal.org/question/qti_v2p0/imsqti_oviewv2p0.html

[17] IEEE. Draft Standard for Learning Object Metadata. Retrieved September 27, 2009, from http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf

[18] R. Jones and T. Boyle. Learning Object Patterns. *Interdisciplinary Journal of Knowledge and Learning Objects*, Vol. 3, 2007

[19] R.N. Katz. Open to Change: An Interview with Leaders of the Open University, *EDUCAUSE review*, 58-64, Mar/Apr 2008

[20] R. Koper and C. Tattersall (Eds.). *Learning Design: A handbook on Modelling and Delivering Networked Education and Training*. Springer 2005

[21] B. J. Krämer and H.-W. Schmidt. Components and Tools for On-line Education. *European Journal of Education*, 36(2):195-222, 2001

[22] B. J. Krämer and A. Zobel. Rollout of CampusContent – DFG-Competence Centre for elearning, *e-learning & education*, July 2008. Retrieved September 27, 2009, from http://eleed.campussource.de/archive/4/1417/

[23] Liferay: Enterprise Open Source Portal. Retrieved September 27, 2009, from http://www.liferay.com/web/guest/home

[24] metacoon: Learning Management System and Authoring Tools. Retrieved September 27, 2009, from http://www.campussource.de/org/software/metacoon/

[25] H.-J. Mittag. Virtual learning environments for statistics education and applications for official statistics. Biannual Meeting of the Korean Statistical Society, Seoul, 5 6 November 2004

[26] H.-J. Mittag. New Statitics: Interactive experiments. Retrieved September 30, 2009, from http://www.fernuni-hagen.de/newstatistics/applets/appletIndex_en.htm

[27] Moodle: Course management System. Retrieved September 27, 2009, from http://moodle.org/

[28] OECD, *Giving Knowledge for Free. The Emergence of Open Educational Resources*. Retrieved September 27, 2009, from http://213.253.134.43/oecd/pdfs/browseit/9607041E.pdf

[29] O. Peters, Hinweise zu einem allgemeinen didaktischen Konzept für die Entwicklung von Studienmaterialien im Fernstudium, *ZIFF-Papier 1* 1975 (in German). Retrieved September 27, 2009, from http://www.fernuni-hagen.de/ZIFF/ZP_001.pdf

[30] D.A. Wiley (Ed.). *Open Publication License*, 1999. Retrieved September 27, 2009, from http://opencontent.org/openpub/

[31] D.A. Wiley. Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. In Wiley, D. A. (Ed.), The Instructional Use of Learning Objects: Online Version. Retrieved September 27, 2009, from http://reusability.org/read/chapters/wiley.doc