A New Approach Integrating Modeling and Simulation into an e-Learning Environment of Engineering Study Programs

Dietmar P.F. Moeller
University of Hamburg, Hamburg, Germany
dietmar.moeller@informatik.uni-hamburg.de

Hamid Vakilzadian
University of Nebraska-Lincoln, Lincoln, Nebraska, USA
hvakilzadian@unl.edu

***Abstract**: This paper presents an integrative concept for information and communication technology (ICT) supported education in modeling and simulation (M&S) as part of engineering study programs. The implementation of the M&S program uses ICT as an indispensable part of the modern education system. This technology provides a new teaching paradigm, strategies, challenges, and prospects for active learning. The subjects introduced through e-Learning provide an active learning environment where computers are used as a front end. M&S is used to find the optimized solution, do sensitivity analysis, or validate an idea. Assessment of student performance is also embedded in the e-Learning modules.*

1. Introduction

With advances in information and communication technology (ICT); the development and expansion of the internet; the increase in the number of ICT applications useful for preparing educational materials; the availability of video, audio, and animation processing software; and the declining price of computers and other accessories, the role of ICT in education has been enlarged in many countries. In the last two decades, computer-based learning, in its various forms and versions, opened up the world of knowledge to everyone; and its most powerful variant, the e-Learning platform, has become an instrument that promises big achievements, especially in education programs. E-Learning is used as a general term to refer to computer-enhanced learning (Holmes and Gardner, 2006). It may include the use of web-based teaching materials and hypermedia, multimedia CD-ROMs or web sites, discussion boards and forums, computer-aided assessments, or a combination of all of these methods.

In engineering education, e-Learning has begun to play an important role in different forms and at various levels. At present, there are many universities which offer online courses in all fields of engineering. The Massachusetts Institute of Technology (MIT) and the Indian NPTEL are probably leading the effort in this regard as they put the learning materials and syllabi of many engineering courses online for anyone to use. Although not as prevalent as in mainstream education, research on the use of e-Learning in engineering is also conducted by different authors. This includes the works of Peterson and Feisel (2004), Gudimetla and Mahalinga (2006), and Zou (2007), who studied the feasibility and effectiveness of e-Learning in different disciplines of engineering and recommended project-specific organizational goals and benchmarks. Moreover, in the USA, more than 90% of all post-secondary institutions offer some variety of e-Learning; and online enrolments increased at a rate of 18% (Allen and Seaman, 2005). Over the last six years, the Australian government has invested over $95 million to enhance e-Learning in the vocational education and training sector (Choy, 2007). In China, the government has established the China Education and Research Network with an objective of establishing a nationwide education and research network infrastructure to support education and research in universities, institutes, and schools (Ariwa and Li, 2005). India has also been attracted by e-Learning and the psychological comfort, and promising beginnings encouraged the government to launch a project called the National Program on Technology Enhanced Learning (NPTEL) that offers e-Learning courses for university students. But none of these have introduced a technology-enhanced e-Learning program which embeds modeling and simulation.

However, in contrast to the increase in the utilization of e-Learning all over the world, designing e-Learning strategies, developing e-Learning content, implementing e-Learning courseware, and evaluating e-Learning outcomes all pose certain difficulties and serious challenges. Developing good courseware in engineering by automating the design and construction processes is normally a complex task. The reason for this is the lack of certain frameworks that are able to deliver true dynamic learning products.

Modeling and simulation (M&S) is a methodology which uses insight gained from dynamic processes in engineering systems. M&S is an iterative process that consists of building a model and its simulation using computer-assisted tools. The structure of the model and/or its parameters can be changed in an attempt to match the real dynamic system under test. When building a model, the modeler needs to use various types of information and always keep in mind [Moeller 2000]:

- Goals and purposes of building a model, model boundaries, relevance of the components that are used in model, and level of detail needed for building a model.
- A priori knowledge of the dynamic system being modeled.
- Experimental data consisting of the measurements of the dynamic system inputs and outputs.
- Estimations of nonmeasurable data as well as state space variables of the real dynamic system.

With respect to the spectrum of available models, a variety of conceptual and mathematical representations exist. However, the choice of an appropriate model depends on the goals and purposes for which the model is intended, the extent a priori knowledge is available and the amount of data collected through experimentation and/or measurements of the real system. From a more general point of view, two major facts are important when modeling complex dynamic systems (Moeller, 2000):

- A model always is a simplification of reality, but should never be so simple that its answers are not correct.
- A model has to be simple to allow for easy use.

Moreover, two important boundary conditions need to be taken into account: model building is a compromise between model goodness (exactness of the results obtained from the model) and expenditures for the modeling (the cost of developing the model, its implementation on the computer, and its simulation). Therefore, there is no reason to develop a detailed and expensive model when the increment of goodness is less than the increase in the cost. This point is important because a model is a very compact way of describing an engineering system.

Introducing e-Learning methodology into the field of modeling and simulation in engineering study programs can be accomplished by combining distributed and virtual environments. The purpose is to facilitate the collaboration among local and/or distant students, instructors, and research team members independent of location (space) and time (Bergstedt et. Al. 2003, Thorne 2003, Bersin 2004).

2. A Concept for a Model Server to Integrate Simulation Models into e-Learning Systems

Given the growing number of e-Learning platforms together with access to the corresponding authoring tools to develop appropriate content for e-Learning systems one must ask, "Why not embed modeling and simulation into e-Learning?" From an e-Learning perspective, simulation is one of the most powerful elements of a stimulating learning process. Moreover, integrating simulation models and using additional, didactically elaborated software tools (which help to explain the complex interdependencies within the model, its parameterization, and, after the simulation run, the interpretation of the results) becomes very attractive. However, the potential that lies in this integration can only be achieved if both systems are flexible and couple the simulation and the e-Learning environment based on the following assumptions (Bach et al 2003):

1. Most e-Learning systems are browser based. The access to simulation models has to follow this precondition. This demands the nearly complete independence of the hardware and software used on the client side.
2. The implementation of the simulation model has to achieve a certain standard of quality. A model which is maintained on a dedicated server by a responsible group of experts can keep this quality level at its best.
3. Authors of the e-Learning content must have a special level of access to a simulation model e-Learningin order to design the learner interface according to the context and the level of detail necessary.
4. There must be multiuser access to the model and user-specific administration of simulation runs and results.

Therefore, the possible integration of simulation into e-Learning developed as part of an NSF-granted USE-ICE project will be discussed. The advantages and disadvantages are exposed, and the preferred solution is described in detail along with the results of an evaluation.

The problem of integrating simulations into a web-based learning environment can be handled in different manner. A boundary feature is the platform independence of the resulting implementation. Otherwise a coherent "look and feel" to run the simulations within different client and network platforms cannot be achieved. In consideration of these preconditions, there are four typical major topics to be addressed:

1. Reprogramming of the Simulation Model—The most challenging solution is to reprogram the simulation in a platform-independent programming language like Java. But this approach has some disadvantages. Firstly, the simulation expert is normally not a Java expert per se, which requires a knowledge transfer to enable a Java programmer to implement a simulation. This transfer is time intensive and expensive. Assuming that the programmer executes the reprogramming task, there is a high probability that the results of the new Java program will differ from the output results of the original simulation program. If the actual task was "only" porting of an existing simulation program (representing/including the model) to another programming language (e.g. Java), there is the runtime environment to be taken into account as a potential source of errors (rounding errors, etc.). If the task was the complete reimplementation of a simulation mode, there are even more potential sources of error. Failure in reprogramming large simulations is, in practice, almost guaranteed. The possible sources of errors are almost limitless; and managing such a project is difficult, because no one can forecast the required effort. A further technical disadvantage is that the time needed for running the model depends on the performance of the learner's client computer. Hence, a complex model might lead to unacceptable wait times. The advantage of this solution is that the programmer has full control of the user interface allowing him/her to create different views for different user groups.

2. Remodeling Followed by an Automatic Code Generation—Another solution to the problem is to remodel the simulation in a high level simulation language that can be used as the basis for a subsequent automatic Java code generation, which might be the choice for a simulation expert who wants to create a new model, even if some of the disadvantages of the first solution still remain (e.g., the problem of high CPU load). For a reimplementation of an existing model, almost all disadvantages of the first solution apply to this solution as well. Hence, the only advantage to this solution is the creation of multiple front ends.
3. Remote Access to the Simulation Model Using a Terminal Client—The simplest solution is the use of a terminal client program to remotely access the simulation model. In this case, the simulation is running on a central server. A terminal server program is running on the server that provides an interface for the simulation to be accessed over a network. There are different products available, some that provide web-based access, others that need a special client program. The conjunctive element between these programs is the fact that the user works with the original interface of the simulation. The advantage is that the user receives the original functionality of the simulation. The disadvantage is the lack of didactical preparation of the data for the user. An author of an e-Learning system is not able to edit the user interface or give different access to one model for different users. The centralized solution has the advantage of uncoupling the runtime behaviour of the simulation from the performance of the learner's client computer, with the exception of the network speed. The network load created by the access via a terminal client program may be heavy; hence, the user needs broadband network access.
4. Embedding the Model into a Client/Server Architecture—The integration of a simulation within a distributed client/server architecture offers a tremendous variety of mechanisms to affect the simulation and its representation. Embedding the simulation in a larger software system creates an additional layer which enables the simulation expert, as well as the author, to separate the model description from the input configuration and from the handling of the simulation results. On the client side, the middle tier layer is used to separate the server side calculation from the actual (and normally more simple) presentation of the results. The author can specify different access levels for different user groups, and the author gets extended access to the model as compared to the learner's access. An author can select variables that the learner can change later on. He/she can preset values to variables and set value ranges for variables. Due to this ability, the learner can experiment with the model in a preconfigured way and integrate his/her own experiences into the model, but the didactical correctness of the model representations is controlled by the author. Embedding the model on the server side means little knowledge about the model is necessary, because the model and its runtime environment are viewed as a black box. Therefore, there is no reason to change anything in the model and, even better, no need for some kind of a reimplementation. The output data of the model is stored and can be used independently of the underlying simulation model. This enables the author to create similar user interfaces for different models or even different simulation systems. The learner doesn't have to reorient for each model presented. The advantage is a better acceptance by the learner. Another feature is to create different views for the same model. In this case, a user can choose between these views and get a better understanding.

A good overview of the necessary systems functionality can be achieved by determining which groups will use the system and how. The user groups can be divided into four superior groups. These groups are called actors and are shown in Figure 1.

2.1 Actors Achieving System Functionality

2.1.1 Modeller

The modeller is a simulation expert who provides models for integration into the e-Learning system. These might be small units running within a simulation environment as well as completely new simulations. The modeller is responsible for the correctness of the model. He/she delivers a brief description of the model available for the author as well as the system provider.

2.1.2 Author

An author provides the e-Learning material and integrates a simulation model as a special kind of e-Learning material into a specific e-Learning system. He/she has to embed the right user interface and decide which parameters the user should see and which ones the user should be able to influence. All settings might be viewed as templates. To create these templates, the author has to be aware of the learner's background as well as the desired degree of difficulty. For an easier approach, the e-Learning system may offer a variety of ready made templates which match for standard purposes. In a later step of development, it is planned that the templates might be generated automatically by the system, on the basis of the data collected about the learner and his/her interactions with the learning system. To achieve this feature, the data accumulated about the learner and the system have to be collected in much more detail than in existing systems.

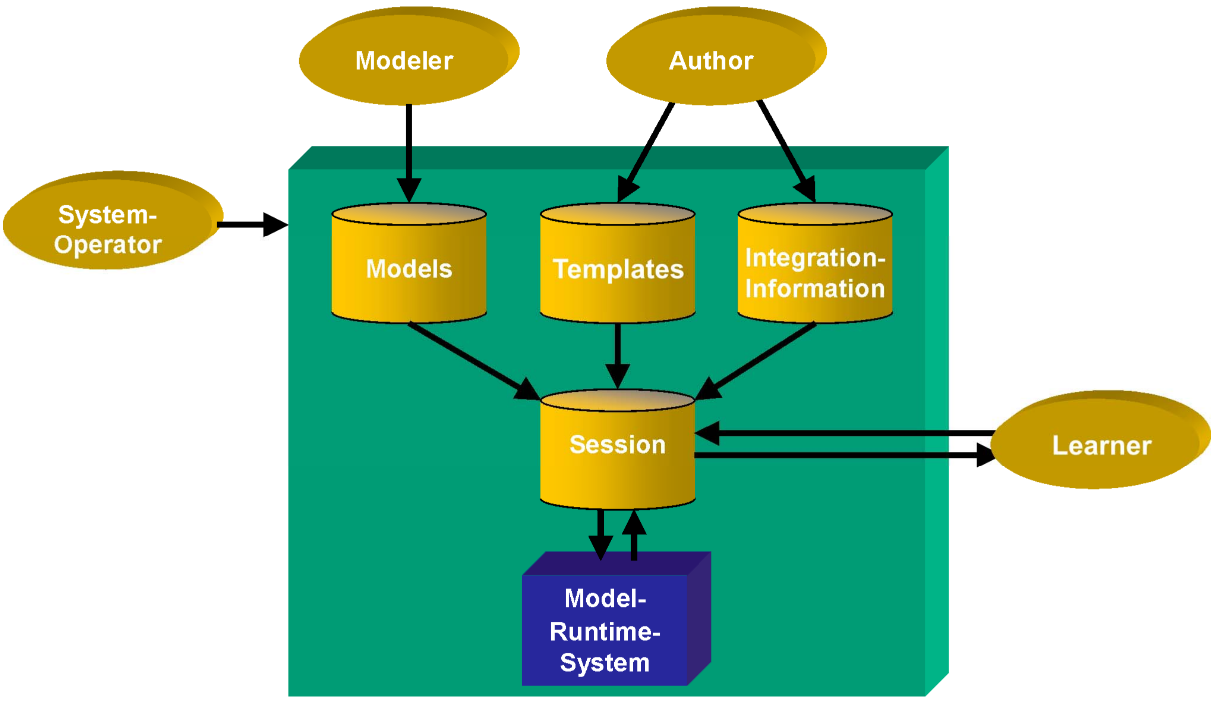

Fig. 1. System functionality described by actors**.**

2.1.3 Learner

The learner has a direct access to the model of the e-Learning system with which he/she is currently working. The simulation system is integrated into the e-Learning system, and the learner doesn't have to turn his/her attention away from the e-Learning system to run a simulation. To optimize the e-Learning process, the simulation has to be integrated in the learning process by adapting it to the surrounding e-Learning system and minimizing the optical and operational differences. The benefit to the e-Learning process, and thus to the learner, is the possibility of experimenting with the model. The learner can perform individualized simulation runs. The positive effect on the learning progress has been shown by empirical studies. Adding a simulation to an e-Learning system is an explorative learning approach which follows the paradigm of problem-based learning. The effect of explorative learning on the learning progress is still being discussed in the literature.

2.1.4 System Provider

A more organizational actor is the system provider, the administration of a server on which the simulation models are situated and running and who provides the software for the different actors. The system provider is responsible for the maintenance and quality assurance of the whole system and the simulation models in particular.He/she Therefore, this role requires a simulation expert. In addition, the system provider has to keep the software tools up to date for the different actors. Furthermore, he/she has to offer technical support and answer questions.

3. Flexibility Concerning e-Learning and Simulation Systems

The flexibility concerning e-Learning systems can be expressed by the following demands (Wittmann & Moeller 2003):

Demand 1: Flexibility concerning the background from which the learner comes.

Course material should be adaptable to learners coming from different application areas by using their terminology, by organizing and labelling data in the way they are comfortable with, and by using examples coming from their own field of interest. It is evident that this demand relates to the way simulation models are offered to the learners as well.

Demand 2: Flexibility in page sequence.

e-Learning material often is very elaborate but very inflexible concerning the sequence a learner has to follow. Naturally, this leads to a lack of flexibility if a course has to be applied to a new application domain or will only be partially used. In these cases, the content has to be copied and restructured afterwards. This functionality is possible for most e-Learning systems; the question, however, is whether the restructured pages maintain their consistency in their content. Often the content has to be rewritten to avoid inconsistencies and false or missing links which result in the demand: a modular content management on the page level is necessary, allowing the opportunity to arrange the content in the sense of a real kit without any redefinition of the relations and interdependencies between the page contents.

Demand 3: *Flexibility of learning level in accordance to the learning path.*

One of the most tedious situations for learners is that the e-Learning material is not well adapted to the level of knowledge of the learner. If the content is too easy, the learners will become bored, and their concentration on the page will decrease. If the content is too difficult, the learner does not feel enough and will be demotivated as well. Therefore, the demand is to provide flexibility concerning the level at which learners enter the system and to adapt the standards and the severity of the content dynamically in dependence on the interactions and the progress during the session.

Demand 4: Flexibility concerning the assembly of the content pages.

The fourth level of flexibility reflects on the course of an e-Learning session and the interactions between the learner and the CBT system. Often, some content is read and wrought repeatedly (e.g., it is obvious to make an example before the theory and to replicate it after the theory is understood) or the user is redirected to the material on special content sequences. In all these cases, it is better for the learner to find some material and content which is different than the example and/or the data he/she has seen on the last page shown. If the system always shows the same example or the same data, the learner will not be able to understand the common link between the example data and will not find the sense of the examples. Therefore Demand 4 postulates a dynamic and automatic change of page content in relation to the position of the page call within the course of the e-Learning session.

Demand 5: Flexibility in the sense of exploratory learning.

A completely new aspect on the structure of a CBT system is introduced by Demand 5. Didactical investigations show that promising learning results are achieved by the concept of exploratory learning. This term means that the system gives the learner stimulation to work through the content by his/her own exploration on an individual path. This idea is explained in much more detail in the papers of R. Schulmeister and cannot be treated in depth here. For the architecture of a learning and authoring system, however, the demand for the possibility to achieve an exploratory learning process leads to the flexibility of level five. It extends the fourth, which concentrates on the modular exchange of content elements on a page during a session to the flexibility concerning even a single page call: the elements on a page should be flexibly linked to each other to facilitate user interaction and exploratory learning.

The flexibility concerning simulation systems can be expressed by the following respective demands (Wittmann & Moeller 2003):

Demand 1: *Flexibility under modeling aspects.*

The development of models should be guided by the users' demand, by the range to which the model will be applied, by the intention behind the model, and the problem which should be solved with the model. The process of limiting the breadth of real-world dynamics to a modeled system and to abstract from real-world details to a manageable model is inherent for simulation as a method for problem solving. In the context of e-Learning, however, the strictly reduced models which answer only one problem are less suitable because the learners are highly differentiated, coming from different thematic areas, having different experiences and previous knowledge, and even having different learning objectives. Normally, the specialized models are not suited to the individualized e-Learning process. There are two technical possibilities to resolve this conflict. The first would be to offer the existing models as they are which leads to decreasing acceptance because the relation to the individual e-Learning situation is not obvious at all. The second would be to adapt the existing models to the current demands of the learner during the current step in his/her learning path. Under didactical aspects, this is the only way a model-based e-Learning environment should be done to keep the process of model building and model maintenance concise and economic. The only solution is modular-hierarchical models not only concerning the breadth of the application but concerning the level of abstraction. Only if the needs and the very individual preconditions and objectives of different learners are previewed during the model building and integration of simulation models into learning systems can the system succeed. The flexibility of such a model tool kit is a precondition for the individually adapted, didactically appealing, and reasonably expensive usage of simulation models within learning environments.

Demand 2: Flexibility under software aspects.

E-Learning normally implies web-based hardware and software platforms. On the other hand, serious models are implemented as more or less closed main programs or they run within a special simulation system. Known solutions try to solve this conflict between flexibility and a proprietary software solution by reimplementing existing models under Java or by offering the original user interface of the simulation via the internet. The disadvantages are, in short: firstly, the lack of efficiency and the tendency to an error-prone reimplementation; secondly, the need for a user who is experienced in the simulation system; and, thirdly, the missing feedback between simulation and learning environment for integrated didactical support. The demand for flexibility, therefore, means a software architecture which is capable of handling those roles and rights which are known from the field of editing learning material, such as authors for learning content, editorial journalists for quality management, learners with their individual learning paths, tutors, learning groups, etc. This range of functionality, which focuses on a very dynamic selection and presentation of e-Learning content, will not be achieved by the architectures of the existing e-Learning systems.

Demand 3: Flexibility concerning the user interface.

The model and its use should be integrated into the e-Learning process, and the model interface must be adaptable to the demands of the learning environment. This includes a differentiated representation of model description (i.e., specification of the dynamic behavior of the system), of the experimental frame (i.e., simulated time interval, accuracy, observation method…), and of the simulation results (final and interim model states, time series, graphs). The content author should be able to select certain clippings from a complex simulation study, encapsulate them according to the didactical target he/she pursues, and offer only this filtered view to the learner within the learning material presented.

4. Application of the Architectural Concept

The architectural concept of an e-Learning system developed as part of the USE-ICE project is shown in Figure 2, which depicts the interrelationships between users, objects, and program modules. The explanation follows the classes of users and their interactions and concentrates on the generation and display of learned modules. The functionalities and common learning environments do not influence the flexibility demands mentioned above. Chat and user administration can be added easily. They are not discussed in the figure. However, the functionalities shown seem to be the most demanding part of the system. In this range of functionality, the new architectures differs most from the existing learning environments.

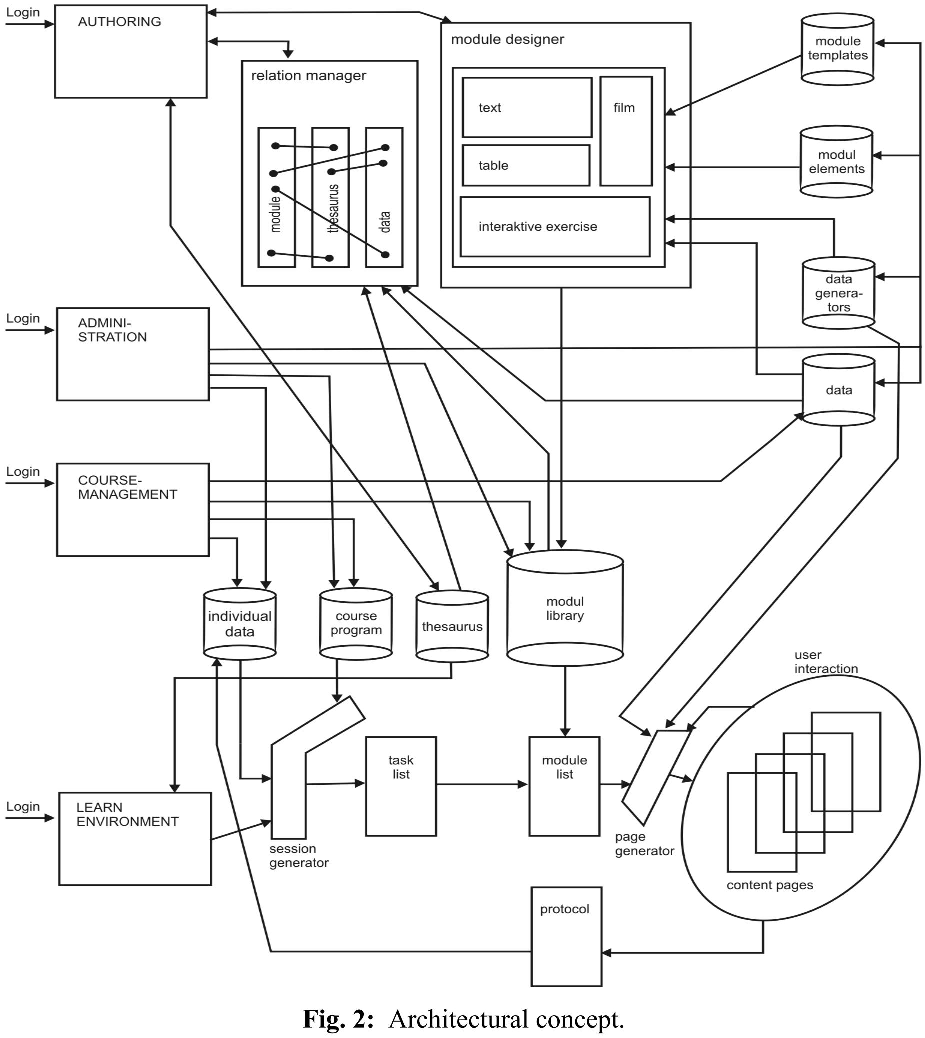

Fig. 2: Architectural concept.

4.1 User Classes

The four different user classes which represent the persons working with the system are:

1. *Learning individuals*: registered for one or more courses and working within the learning system.
2. *Teachers*: managing a course, its content, and its organization.

3. *Authors*: individuals who write new content in the form of new learning modules and who relate the new material to already existing content modules.
4. *System administrators*: individuals who have complete access to all of the data and objects stored in the system to maintain them and to fix technical problems.

System administrators have full access to all data stored in the system. However, if the learning environment is being used (operating), they will not have to interfere. This is why this class needs no further explanation in this context.

Teachers have two roles: administrative functions and online and/or off-line contact for the learners. From the view of software technology, the administrative work (e.g., authoring a course from already existing modules, creating and maintaining a list of participants of the course) will be supported by a database system. This part of the system is very conventional and does not differ from existing learning environments. The same observation can be made for the interactions with the other individuals registered by the learning system. The corresponding software modules can be found ready made and can be integrated easily without any complication in relation to the content management system.

For the user classes, teachers, and learners, the steps during a session with the new system are:

1. The learner enters the learning environment, which offers features such as chat, forums, etc., and starts a learning session. To perform this functionality, a program module called a session generator generates a task list with the remaining topics to be covered. During this process, individual data on the learner, such as learning level and/or recent progress within the course, are taken into account. The task list gives the content and the learning target for the learning session in an abstract form, which has to be transformed into a set of learning modules of the learning system.
2. These learning modules are highly parameterized dynamically and, therefore, have to be filled dynamically by adding the text modules, pictures, and examples which correspond to the learning progress, the background, and the course into which the learner is logged. This is the task of the so called page generator.

This program module either generates the missing information directly from the database or generates it using an algorithm represented by a data generator module (e.g., a sequence of random numbers according to a given distribution by a random number generator). Afterwards, a set of fully parameterized pages which consistently relate to each other is available for processing during the session. However, this flexibility requires adaptability caused by user interactions. For example, the page generator will have to be recalled to offer changed data sets within a repeatedly called example, or an exercise has to be updated due to the completed learning level. Therefore, all interactions and learning steps in the protocol lead to online or off-line evaluation; and they are added to the individual data set of the learner. The most demanding task for the system is the way new content can be integrated.

In the terms of the system architecture, the question that must be answered is how the module library can be augmented with new content. The user class allowed to do this is the authors. Authors get a special user interface to interact with the system and are supported by two quite independent software components which focus on editing the modules, the module designer, and on linking the new edited modules to existing material and also to the relation manager.

4.2 Module Designer

The module designer is an editor tailored to the needs of editing learning modules for the learning system. It is based on a set of given templates and element types to fill the pages. The use of templates has two purposes: the resulting content should be displayed in a uniform layout, and (this is the main difference between this system and others) the templates should imply semantic restrictions that the author has to hold. For example, the description for the structure of an interactive simulation example cannot be edited using the template for explaining text.

The restriction on a given set of elements and on the content pages should be the type and the usage of a media element that is used on a content page. The system can achieve a proper display and an automatic substitution if content is changed dynamically. The media elements used on the content pages are texts, pictures, films, graphs, formulas, tables, and so on. A special class of elements are the data elements. These elements can be filled either by access to the database of the system where the example data lies or by a data generator which generates dynamically the data to display at the time of the page call.

Thus, the new material can be edited and persistently brought into the systems module database as a new media element or as a new content module using the media elements within a semantic restricted module template.

4.3 Relation Manager

Even if the module is completed, it has to be linked to the other modules of the kit semantically. This task has to be done manually by the author. However, it can be supported by the Relation Manager. This program module gives the author the opportunity to define relations between:

- New and existing modules
- New and existing modules and their media elements
- New and existing media elements without their representation in a particular module
- Module content and the thesaurus of the system
- Module and media elements to courses
- Module and media content to application areas

This set of relations is extendable by the authors and, thus, can be individually adapted to cover special semantic restrictions and the context and/or background module rules. The specification and the administration of the relations are done graphically by linking graphically represented objects using labelled lines. Examples of possible relations could be: “is subordinate term to,” “contains example to,” “needs data from,” “is explained in,” etc. The huge set of relations can be managed concisely by restricting and filtering the object classes and the relations involved. Thus, despite the enormous amount of relations that the system holds, the author can work with a concise subset and does the important work of linking the content elements semantically with justifiable effort.

To fulfil these demands a client/server architecture was selected after evaluating the different possible approaches. Figure 3 shows the resulting architecture of the client/server-based system with the central server on one side and the different clients on the other. The different actors involved in the procedure have different needs and, hence, need different clients.

The server is the centralized location for storing and retrieving data. In advance of the usual data management, the server, in this case, has to store simulation model descriptions and has to support the interfaces according to the model runtime environments. For each model, the desired configuration and parameterization has to be selected before each run. The simulation models are encapsulated in services as well. At this point, the simulation model and its runtime environment are viewed as a black box. The input for the models is generated using the data from the database and the actual (and validated) user input. The data that informs the client which parameters are modifiable and in what way is transmitted by the server in an early step in the client/server communication protocol. After the run, the simulation results are collected and processed by corresponding rules and conditions of the surrounding service. In this step, the simulation results are separated from the model description and are presentable in completely different ways later on. The transformed simulation results are transmitted to the client using TCP/IP. The data structure used for input and output parameters is tree based and extensible. Thus, the data representation is very flexible. The final presentation sequence included in the tree structure may contain, for example, graphical settings like scaling and colours, titles, or audio files.

The tasks occurring on the client side are very divergent. Thus, different clients are designed for the different actors. The client for the learner has the main task of (graphically) presenting the simulation results. Various modules have different needs for the presentation of the results. One applet might not be able to meet all of the demands. To support the system provider in implementing new applets, a framework is designed that provides fundamental functions and can easily be modified or expanded. The author can choose a ready-made applet for the presentation of the simulation data or ask the system provider to adapt one for his/her needs.

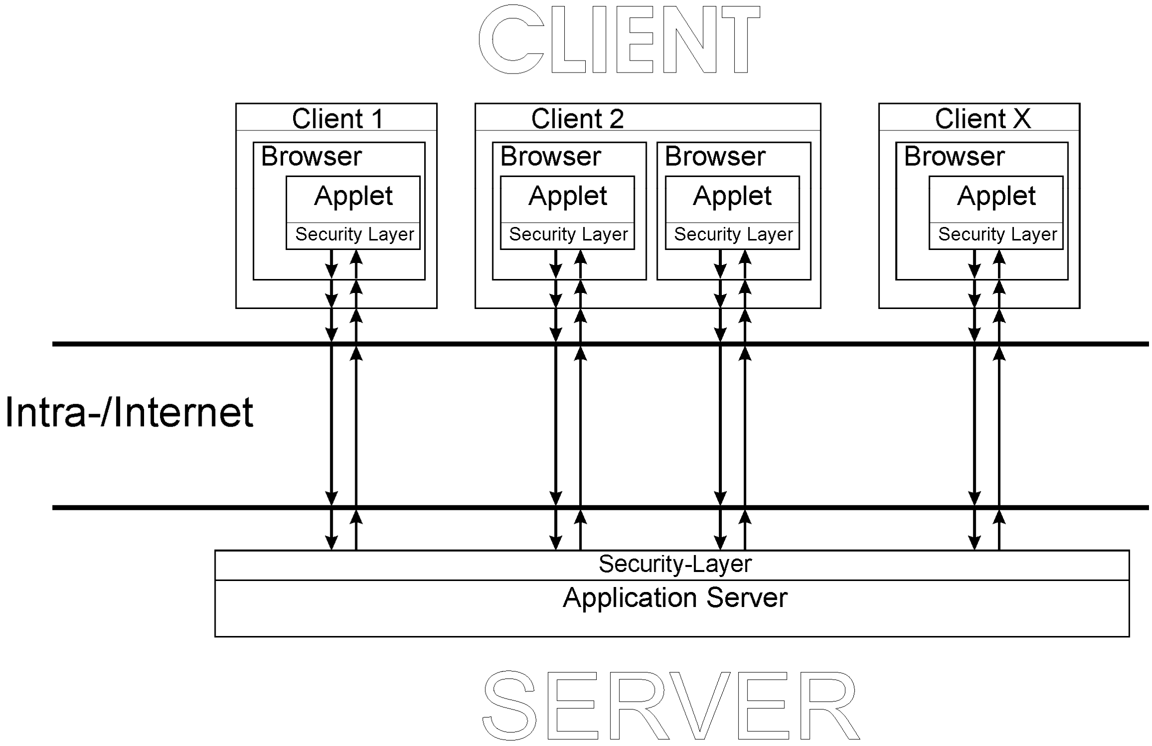

Fig.3: Client/server architecture.

5. Conclusion

The system concept explained in this paper concludes with the following assumptions:

1. Building a reusable e-Learning system efficiently requires that the demand for flexibility must be considered.
2. Existing systems fulfil these demands only partially. There is a lack of consistency between author systems and e-Learning environments. Therefore, the users reach flexibility by explicitly programming content pages which are hardly reusable.
3. Obtaining flexibility requires the internal structure of a content page to be taken into account and the relations between the found media elements to be specified and administrated.
4. The system concept proposes an architecture which is based on the described media elements which are static and represent the content of a page. The user-defined relations, which are administered by a special graphical program module within the system, are mapped dynamically.
5. To do so, a comprehensive architecture for an authoring, learning, and content management system is proposed. This can be easily extended to standard features to provide a solution for efficient e-Learning system usage.
6. The integration of e-Learning and simulation is realized by a client/server architecture, shown in Figure 3, to achieve a flexible system solution.
7. The e-Learning material is embedded into the free, available Moodle platform, coupled with the most common simulators for the different engineering domains, which are pSpice, Matlab Simulink, ModelMaker, Comsol Multiphysics, etc.
8. The e-Learning modules developed thus far concentrate on the following topics:
   - Introduction to Systems
   - Introduction to Modeling and Simulation

- Concepts in Continuous Time Simulation
- Mathematical Description of Continuous Systems
- Concepts in Discrete System Simulation
- Statistical Models and Their use in Simulation
- Verification and Validation of Simulation Models
- Monnte Carlo Simulation and its Applications
  Case Studies

Acknowledgements

This work is supported in part by the National Science Foundation, Directorate for Education and Human Resources (EHR), Division of Undergraduate Education (DUE), under Grant Number CCLI-0737530. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

Holmes B., Gardner J. (2006). e-Learning; concepts and practice. Sage Publications, Los Angeles, 186 pp.

Allen, E., Seaman, J. (2005). Growing by Degrees; Online education in the United States, Sloan Consortium.

Choy S. (2007). Benefits of e-Learning Benchmarks: Australian Case Studies. The Electronic Journal of e-Learning, 5(1): 11 – 20.

Ariwa E., Li R. (2005). The impact of e-Learning on China Education and Research Network (CERNET). Proc. of the 2nd Int. Conf. on e-Learning for Knowledge-Based Society, August 4-7, Bangkok, Thailand, 28.1 – 28.7.

Peterson and Feisel (2002) A Colloquy on Learning Objectives For Engineering Education Laboratories. ASEE Annual Conference & Exposition: Vive L'ingenieur!, 16-19 June.2002, Montreal Canada.

Gudimetla and Mahalinga (2006). The role for E-Learning in engineering education: creating quality support structures to complement traditional leaerning. In: 17th Annual Conference of the Australasian Association for Engineering Education, 10-13 December 2006, Auckland, New Zealand.

Möller, D. P. F. (2000): Perspectives on M&S Education in Science and Engineering, pp. 17-24, In: Proceedings ESS 2000 Simulation in Industry, Ed.: D. P. F. Möller, SCS Publ. House, Ghent,

Bergstedt, S., Wiegreffe, S., Wittmann, J., Möller, D. P. F. (2003): Content Management Systems and E-Learning Systems – A Symbiosis?, In Proceed. 3rd IEEE Internat. Conf. on Advanced Learning Technologies (ICALT), pp. 155-159, Eds.: V. Devedzic, V. Spector, J. M. Sampson, D. G. Kinshuk, IEEE Publ. House, Los Alamitos

K. Thorne, Blended Learning, London, U.K.: Kogan Page, 2003

J. Bersin, The Blended Learning Handbook, New York: Wiley, 2004

Wittmann, J., Moeller, D.P.F. (2003): Demands for flexibility in authoring and e-Learning platforms under the aspect of integrating simulation model. Published in: Proceedings of the “eChallenges 2003”, European Commission, Bologna.

Bach, M., Himstedt, J., Wittmann, J., Moeller, D. P. F. (2003) A concept for a model server to integrate simulation models into web-based learning environments Published in: Troch, I., Breitenecker, F. (Eds.): 4th IMACS Symposium on Mathematical Modelling, Vienna, February 5-7, 2003, Argesim-Verlag, Wien 2003. Vol.2, pages 303-312