Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

A Web-based learning tool for applied mathematics disciplines

Valeria Marina Monetti, Mario Rosario Guarracino,
Francesco Giannino, Loredana Randazzo,
Gerardo Toraldo

RT-ICAR-NA-2005-17 **Ottobre 2005**

Consiglio Nazionale delle Ricerche, Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR) – Sede di Napoli, Via P. Castellino 111, I-80131 Napoli, Tel: +39-0816139508, Fax: +39-0816139531, e-mail: Napoli@icar.cnr.it, URL: *www.na.icar.cnr.it*

Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

A Web-based learning tool for applied mathematics disciplines [1]

Valeria Marina Monetti [a,*], Mario Rosario Guarracino [b],
Francesco Giannino [a], Loredana Randazzo [a],
Gerardo Toraldo [a]

Rapporto Tecnico N.:
RT-ICAR-NA-2005-17

Data:
Ottobre 2005

---

[1] lavoro sottomesso a *Computer and Education*,
[a] *Department of Agricultural Engineering, University of Naples Federico II,*
Via Università 100, 80055, Portici - Napoli (Italy)
[b] *High Performance Computing and Networking Institute Italian National Research Council*
Via P. Castellino, 111, 80131, Napoli (Italy)

A Web-based learning tool for applied mathematics disciplines

Valeria Marina Monetti [a,*], Mario Rosario Guarracino [b], Francesco Giannino [a], Loredana Randazzo [a], Gerardo Toraldo [a]

[a] *Department of Agricultural Engineering, University of Naples Federico II, Via Università 100, 80055, Portici - Napoli (Italy)*

[b] *High Performance Computing and Networking Institute Italian National Research Council Via P. Castellino, 111, 80131, Napoli (Italy)*

Abstract

Recently, many efforts have been devoted to the implementation of web based learning tools, that have found widespread diffusion in high schools, graduate and undergraduate universities classes. They enable the fruition of educational materials through a web used user interface and they often implement the same cognitive model that is beyond a traditional course.

The aim of this work is to present SIRMM (Searchable Information Repository of Mathematical Models), an interactive environment for mathematical teaching and learning within scientific disciplines. It provides a mathematical common and unifying framework to teach scientific and technical disciplines such as physics, engineering, biology and finance, in which mathematical modeling and data analysis play a major role. It enables the development of training paths that are both interactive and multidisciplinary, in which mathematics is used to easier the understanding and analysis of scientific problems, building and solving ad hoc models.

In the present work we describe the conceptual and technological issues of the system and provide an example of its use in a mathematical modeling graduate course.

Keywords: Learning Environments; architectures for educational technology systems; teaching/learning strategies; improving classroom teaching.

1. Introduction

Mathematics plays a major role in scientific education as a basic tool in solving problems, explaining phenomena, developing new ideas and for a better understanding of reality. It is well-established that a modern teaching approach to scientific disciplines needs mathematics as a basic tool for analyzing and solving real problems [1, 2, 3, 4, 5, 6]. In some sense, from the well established and traditional role of "science of numbers and their operations, interrelations, combinations, generalizations, and abstractions and of space configurations and their structure, measurement, transformations, and generalizations (Merriam Webster Dictionary)", Mathematics tends to go back to its original roots of "màtēma", i.e. everything which can be learned from the experience, and therefore the knowledge in a broad sense, i.e. not just logical reasoning and quantitative calculation, but especially what can we can learn from experiencing them in analyzing and solving real problems.

Nowadays, the global market and the technologically advanced world demands for qualified professional skills to use technological and computational tools, to correctly analyze data, to efficiently support sustainable decisions, to test conjectures, and to develop scenarios of complex systems through a process of modeling and synthesis of the real world. In other words, people are required to have a scientific and technical background that allows to fully exploit the potentialities of the widely available Information and Communication Technologies (ICT) resources. It is important to observe that such potentialities dramatically increased in the last ten years, both in terms of "range of the problems" and size of the instances that can be solved with general purpose ICT tools, i.e. tools that belong to the general toolset of most of the computing and communications resources available on educational and working environments today. This is not just because more and more powerful and cheap computers are available, but mainly because user-friendly software tools (interactive computational environments) allow to easily access large data set and computing power.

For instance, a linear optimization problem, coming out from an equilibrium problem or from some economical market planning strategy, with some hundreds of variables can be solved by a high school student with basic mathematical tools and a basic Office Package; the same considerations hold for every applied problem that can be modeled as a

[*] Corresponding author. Tel. +390812539413; fax: +390812539412
E-mail address: vmonetti@unina.it

combinatorial optimization problem. It is important to observe that a large set of real world problems can be represented with such models, without requiring advanced knowledge of differential calculus or theoretical mathematics.

In this context, the advancement of ICT can be of great help also to the development of new teaching models within scientific disciplines, by integrating computer and communication technologies with more traditional learning paths. More specifically, the goal to be achieved through the use of ICT is to create new advanced instructional supports to integrate the basic statistical-mathematical knowledge in a computationally effective way. This learning approach based on the *problem solving* paradigm allows:

- to analyze and understand scientific phenomena by means of models and simulations;
- to make experimental data analysis and systems simulation;
- to use numerical and computational tools.

In this process, computational modeling, numerical analysis, and (high performance) computing strategies are not the focus of the lesson, but rather naturally enter as elements of the problem solving process (De Corte et al., 2000; Poggio, & Smale, 2003; Marchioro II, & Landon, 1997). Such innovative approach needs tools and techniques which are quite different from the classical ones and several components are involved in applied discipline lessons in order to make effective an integrated educational process:

1. Traditional "learning tools", mainly textbooks ranging from the more theoretical and methodological to the computational-oriented.
2. Traditional "computational tools" (numerical methods, algorithms, software) available in books and on the net, whose technical descriptions and documentations are suitable for expert and well trained users
3. Innovative ICT tools, such as databases, web-learning tools, problem solving environments (PSE) (Avgeriou, Retalis, & Skordalakis, 2001; Dede, 1996; Eastmond, & Ziegahn, 1996), which are going either to replace or to integrate the traditional computational tools.

Usually an educational process is based on a blend of several traditional an technologically advanced supports, sharing a large quantity of information but using heterogeneous approaches and languages.

Although a number of innovative projects are rapidly evolving, from the methodological point of view there are still many gaps to be filled in among the different components, previously mentioned, that should create a modern educational process. Despite of the huge amount of educational material which has been made available online (lecture notes, interactive presentations, handbooks, teaching software, computational tools, etc…), the goal of merging all these components in a unifying online educational applied mathematical modeling framework is still far to be accomplished. In particular, to create interactive Learning Environments (LE) based on ICT it would be highly recommended to use automatic systems (Moallem, 2001; Marchioro II, & Landon, 1997; Pohjolainen, Hantakanges, Ranta, Levasma, & Pesonen, 2003; Smith, & Ragan, 1993; Harasim, 1996).

A first meaningful step in such direction are the Learning Technology Systems (LTS) (Avgeriou, Papasalouros, & Retalis, 2001; Rafaeli, Barak, Dan-Gur, & Toch, 2004; Kong, & Kwok, 1998; Sheremetov, & Arenas, 2002) which provide knowledge through Internet and WWW technologies, offering a powerful tool to integrate knowledge and to build interdisciplinary educational paths (Michailidou, & Economides, 2003). "They often provide facilities to adapt to the context, allowing users to change certain system parameters and adapt their behaviour accordingly" (Oppermann, Rashev, & Kinshuk, 1997). Moreover, several websites have been implemented in the last few years with the main purpose to supply a new context for learning and teaching applied mathematic (see for example http://archives.math.utk.edu/calculus/crol.html) and to enhance interactivity between tutors and learners (Lowe, & Hall, 1999; McCormack, & Jones, 1997), providing a large amount of information about mathematics, mathematical software and applied sciences, collections of experimental data (http://lib.stat.cmu.edu/DASL/DataArchive.html; http://mathforum.org/workshops/sum96/data.collections/datalibrary/) and, finally, more or less comprehensive virtual libraries of mathematical software maintained by universities. For instance, ZIB (http://elib.zib.de/) is an electronic library of mathematical software and information that operates in the field of information technology. Galaxy (http://www.einet.net/galaxy/Science/Mathematics.html) is a web site of miscellaneous branches of mathematics that provides an index of pointers to resources which are gathered into categories. CSC (Center for Scientific Computing of Finland: http://www.csc.fi/math_topics/General.html) provides modeling, computing, and information services for universities, research institutions and industrial companies. One of the most interesting projects is GAMS (Guide to Available Mathematical software), a project of the National Institute of Standards and Technology (NIST), whose goal is to provide "a cross-index and virtual repository of mathematical and statistical software components of use in computational science and engineering" with centralized access to such items as abstracts, documentation, and source code; rather than operate a physical repository of its own, this system provides transparent access to multiple repositories operated by others. Moreover "All cataloged problem-solving software modules are assigned one or more problem classifications from a tree-structured taxonomy of mathematical and statistical problems" (more details can be found at http://gams.nist.gov/). Finally, many databases are available to promote the communication and distribution of applied models in specific scientific area; for example, REM (http://dino.wiz.uni-kassel.de/ecobas.html) is a meta-database for existing mathematical models and StatLib (http://lib.stat.cmu.edu) is a system for distributing statistical software, datasets, and information by electronic mail, FTP and HTTP.

From the educational point of view, an increasing and crucial role is going to be played in the future by PSE, which represent general purpose and very powerful tools that allow to deal with the problem solving stage in a painless and rather efficient way without specific knowledge about the code. A PSE environment (Gallopoulos, Houstis, & Rice, 1994; Gallopoulos, & Sameh, 1997) is a "computer system that provides all the computational facilities necessary to solve a target class of problems. This features include advanced solution methods, automatic or semiautomatic selection of solution methods, and ways to easily incorporate novel solution methods. Moreover, PSEs use the language of the target class of problems, so users can run them without specialized knowledge of the underlying computer hardware or software." Their aim is mainly to be an user-friendly software tool, whose computing efficiency is often sacrificed for the easy-to-use features. Recent trends in the scientific literature show an increasing use of PSE environments such as Mathematica, Matlab or Simulink, as support for scientific disciplines in engineering and scientific education. Their languages, much closer to the natural language than the classical programming languages, have been adopted into modern applied mathematics books for problems and algorithms descriptions **(**Quarteroni, & Saleri, 2003**)**.

In conclusion, a large amount of projects/material that is available online, has been planned with the common goal to redesign educational process through the integration of computer and communications technologies. Nevertheless, still two serious drawbacks seem to be the heterogeneity of the different projects and the lack of portability and adaptability to different educational scenarios.

The aim of this work is to present SIRMM (Searchable Information Repository of Mathematical Models), a new interactive environment for mathematical teaching and learning within scientific disciplines. The goal of the SIRMM project is to provide a mathematical common and unifying framework to teach scientific and technical disciplines such as physics, engineering, biology and finance, in which mathematical modeling and data analysis play a major role. On the other way around, SIRMM might be used in teaching and learning mathematics through real world applications. SIRMM is not supposed to be a simple collection of problems, rather it is an adaptable LE to be used in a wide range of courses such as biology, ecology, chemistry, mathematics and physics. It enhances scientific education, enlightening on the role of computer modeling in science. It enables the development of training paths that are both interactive and multidisciplinary, in which mathematics, statistics and computer science, available through the software tools, are used to easier the understanding and analysis of scientific problems, building and solving ad hoc models.

2. SIRMM Conceptual Architecture

SIRMM proposes a conceptual framework and architecture for the development of a problem solving-oriented learning system for applied science disciplines, in which the main mathematical concepts involved in real word applications are supplied in a unified, flexible and collaborative framework. It realizes a link between mathematics and real world, providing easy access to indispensable mathematical notions. It can find application in disciplines such as engineering, biology and economy, in which mathematical modeling has a main role in the comprehension of key facts. It can be used ad different depth levels and with different paths tailored on students' capabilities. Finally it can be integrated with external resources.

General goals of SIRMM are:

- To represent a useful tool, in conjunction with innovative cognitive models applied to mathematic teaching, to integrate different themes and disciplines;
- To stimulate the development of analytic, synthesis, modeling and evaluation capabilities of reality and the creation of multiple scenarios;
- To overcome knowledge separation actually existing among different disciplines, providing a smoother an homogeneous vision of reality;
- To provide a constructive interaction tool among the different agents of knowledge process through a media that foster the exchange between knowledge producers and consumers, in a process of constant feedback on ideas, contents and methodologies.

SIRMM database contains objects of different types usable in frontal and distance learning processes within standard learning paths. SIRMM has the ambitious goal of providing and integrating appropriate and different learning paths for different types of students and subjects field. It implements a learn-by-doing approach, thus yielding the greatest educational benefit. In contrast to traditional science, students are presented with an LE in which they have the option to dynamically create their own study path. Furthermore, students have also the chance of accessing additional information and tools as need arises, stimulating a flow of knowledge between producers and consumers.

SIRMM let people solve problems step by step from the easiest level to the most difficult one. The system provides the possibility of building various modules of information related to every specific issue, and different educational paths; Figure 1 presents two alternative paths. The first one focuses on mathematical issues (calculus and numerical analysis and computational techniques), by integrating application problems in the learning path, whilst in the second one the mathematical concepts support the formalization and the problem solving stage in a technical discipline.

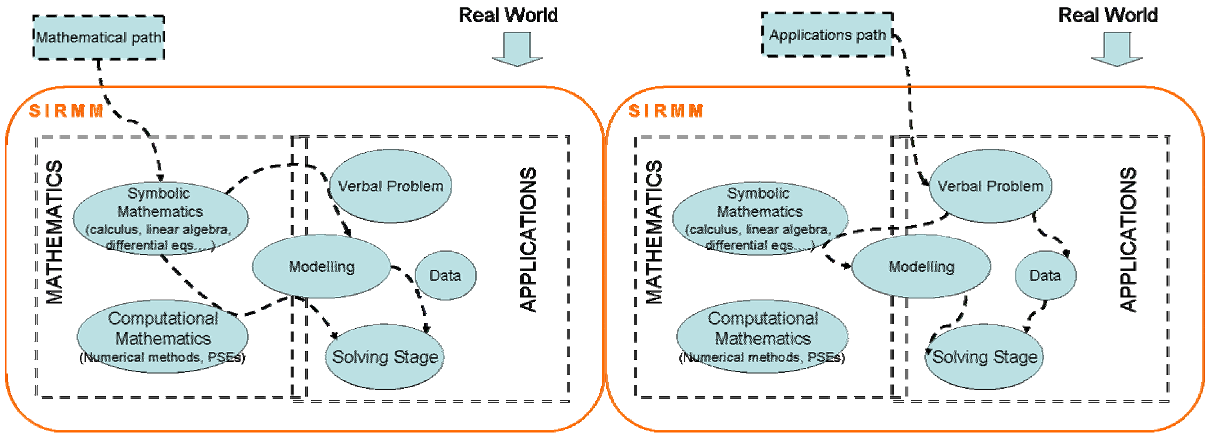

Fig. 1. Example of learning paths supported by SIRMM.

The information granularity behind SIRMM substitutes the traditional structure of knowledge arranged in a rigid form, overcoming knowledge separation and obstacles in multidisciplinary approaches. Users identify and browse the learning objects that characterize the steps of a specific learning path as components which may be reused in the comprehension of other problems. Fine-grained modules are reusable for different courses and disciplines, facing the needs of an educator to teach the desired course and convert a module already developed into another one. SIRMM is flexible enough to address changing user requirements and knowledge, and is usable in a variety of contexts.
Given its flexibility, SIRMM might also provide useful teaching and learning facilities for mathematics courses based on the problem solving paradigm. In a classical course of mathematical studies, the various arguments are presented one after the other in a linear sequence, using what is usually referred as the objectivist model. A modern cognitive approach should permit to illustrate concepts with the aid of practical applications, described in a homogeneous framework. A constructivist educational process might fulfill such requirements. Indeed, following the constructivists theories (Vrasidas, 2000; Dick, 1992; Hannafin, Land, & Oliver, 1999; Jonassen, & Grabowski, 1993), the learner has an active role in the learning process, since learning takes place in a dynamic step by step process, that starts from his experience. New knowledge is added to existing one, through a constructive process in which the learner has an active role, in constant interaction with the environment. The motivations that justify the latter approach are based on the so called learning by doing, which brings motivation and stimulus to the process. This is particularly appropriate with respect to mathematics, which is perceived as a mere collection of formulas. Students construct new knowledge on the basis of the previously acquired, with their own activity in constant interaction with the surrounding LE. In the next sections a solution to issues regarding reusability, adaptability and flexibility are detailed, in order to prove that SIRMM makes it possible to enhance the use of mathematical techniques in scientific and technical disciplines, through the realization of ad hoc learning paths.

1 SIRMM platform

SIRMM is based on an interdisciplinary interactive and multimedia database. It is accessible through a web browser (http://www.sirmm.unina.it) that integrates different scientific oriented applications. It provides, for each problem:

1. a clear description of the phenomena,
2. a model definition (data, relations, parameters) with the analysis of the mathematical issues,
3. a numerical-computational approach,
4. case studies with analysis and interpretation of the results,
5. links to additional source of information.

2 Model database

SIRMM is composed of a set of entities interacting at different levels. In this section a model for those entities and their relationships is given. The design of the learning management system will be derived from this analysis.
Figure 2 provides an Entity-Relationship model. We start describing the entities. `DISCIPLINE` describes the scientific area of interest, that is the fields in which the contents of SIRMM are catalogued. Each discipline includes several `PROBLEM`s, that are identified by a name and a short description. A problem is formulated as a `MODEL` (possibly more than one), composed, according to Von Neumann, of equations and a description in natural language of what it represents in that context. Each `MODEL` is identified by its name, and it can be related to different `PROBLEM`s. The following step concerns the computational solving stage, which involves a `SOFTWARE` which implements a suitable solving method for the `MODEL`, and input `DATA`. Each `SOFTWARE` can be either a code (specifically created for SIRMM) or a link to some external PSE or library. Input `DATA` are given in a format(s) suitable for related

SOFTWARE. A specific instance of a problem can be described in natural language and becomes a CASE STUDY. Each of those entities is related to one or more RESOURCEs (i.e. book, article, …), to provide further reading on the topic.

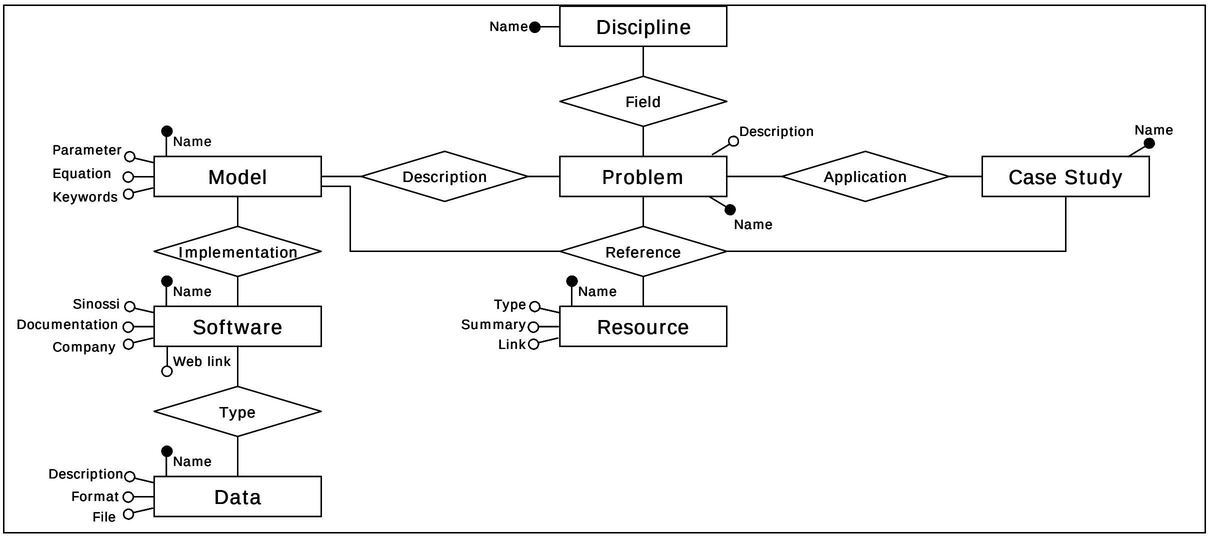

Fig. 2. Entity-Relationship diagram.

The logical model of entities and relations induces a model in the navigation of the information stored in the system. Since each entity is in relation with one or more entities, it is possible to start the navigation from each of them. For example, a student can have some data, and he might want to discover which are the software that can accept them as input. As another example, one could start from a particular problem, then studying the available models. For one model one can be interested in software that implements an algorithm that can solve the model for given data. For each model auxiliary information can be accessed, such as references, slides, books, for a better understanding of the topic. Figure 3 provides an example of population of SIRMM repository and its use. In the field of Ecology, a well known problem is the dynamic of a population. Many well established models describe the dynamics of one species (e.g. Malthus, Logistic) and, among those, the logistic model can be used to introduce the dynamics of multi-species population. Those models are well implemented in system dynamic software tools (e.g. SIMILE). For simulation and testing, real data will be needed. A particular instance of this problem can be found when we decide to study how Roe (Capreolus capreolus) population behaves in a particular context, such as North Italy National Park. For that particular problem (and/or model, software and case study) there are references for further reading.

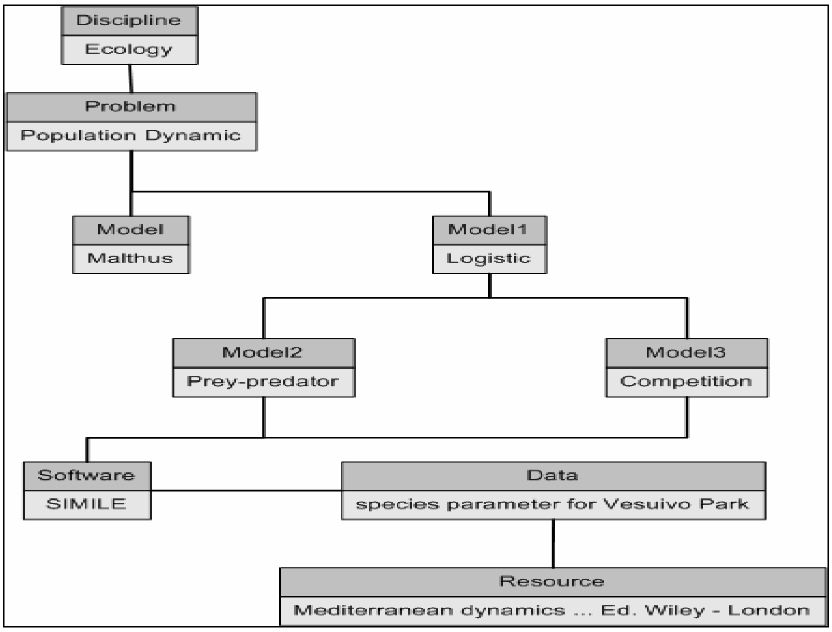

Fig. 3. An example of information representation in SIRMM.

It is worth noting that in SIRMM new links among different information can be added. Indeed, in the system there could be models different from the logistic one, that can be used to model the dynamic population problem, and that have been inserted to solve completely different problems. Among the existing models, if the users identifies one that can be used to solve the problem, this relation can be added. In all successive navigations, users will be able to access this new information. This example shows the capability of the system: each time a new link is identified it can be added to the system so that all users can benefit from the new knowledge. In this way the structure of the information is continuously evolving.

3. SIRMM Technological Architecture

The technological architecture of SIRMM has been developed using the logical model of layers. In this model, system functionalities are divided and implemented in software elements called components. Each component uses the functionalities of those laying in a lower layer and it is used by at least one component in the upper layer. In this way, it is possible to add functionalities at a certain level simply knowing how to use the functionalities of the lower level. Such a software system is said to be open, if it is possible to implement it from the specification of its components interfaces. The advantage of an open layered architecture are implementation modularity, transparent use of lower functionalities, and easy detection of errors or weaknesses. Indeed, if a new functionality needs to be added to the existing ones, it is possible to identify which are the already existing components that can be used for its implementation and therefore the layer in which it has to be collocated, to make it available to upper layers. Further, only the knowledge of interface specification is needed, while it is not required to know how the lower layer components have been implemented. In this way, a physically distributed team can develop the system. Finally, a system error can be tracked starting from the analysis of the components in which the error is present: the error can be only in those components in the lower layer that are used by all upper level components in which the error is present. The layer in which the fault components are located is identified as the lower layer in which there are components that are used by all upper components that are faulty.

The proposed organization of SIRMM software components into layers provides an easy way to abstract and collocate each functionality provided by the software system. A careful architecture specification, at an early stage of the project, has provided reference basis for system implementation. Moreover, it will permit to expand the system in many possible ways, as it is shown in the following. We have chosen each component with the aim to obtain a efficient, scalable and portable system. Efficient means that the system can provide minimum mean waiting time. This is important because users of interactive systems require a fast elaboration of their requests, and waiting time is perceived as a system ineffectiveness. Scalable is the ability of the technological architecture to continue to function smoothly as additional data volume, users and elaboration is required. Finally, a software system is portable if it can be moved to a new operating system and hardware without having to make any changes related to its implementation.

In this section a detailed description is given of the various layers, the components present in each layer and their functionality. This approach represents a de facto standard in open system software prototyping and implementation in the field, as it is outlined, for example, in (Avgeriou, 2001b).

All components are logically divided in the following four layers:

- System software layer – It offers components managing basic services provided by operating system, which are not in its kernel.
- Middleware – It contains all system services that are not readily available in the operating system of the server hosting the repository and that are used to deliver it on the internet.
- Application general – It comprises all components that are not specific to SIRMM and that can be re-used by future applications with similar demands.
- Application specific – It encloses all software components that have been specifically implemented for the application and that can be used by other systems with the same needs.

To describe the system, we use the UML graph for software system description and we will give a detailed description of the functionalities of each component. Figure 4 completely describes the architecture: it identifies its components, it shows their position in the layers, and the dependences existing among them in the various layers. The software reference architecture has been firstly proposed in (Giannino, Guarracino, Monetti, Romano, & Toraldo, 2004a; Giannino, Guarracino, Monetti, Romano, & Toraldo, 2004b) and here it is discussed in view of the prototype implementation we have developed.

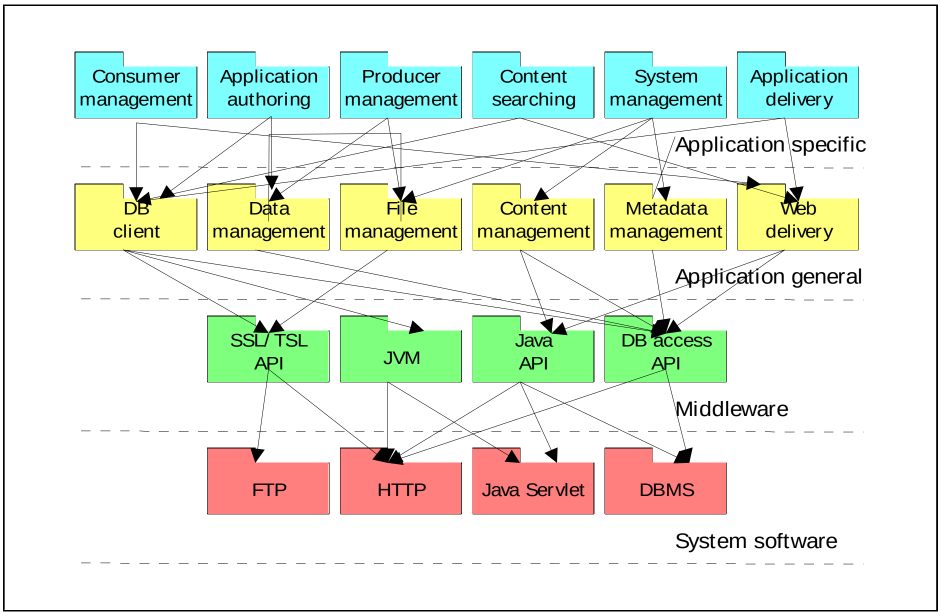

Fig. 4. Software architecture.

System software layer contains the components that implement network protocols, relational database management system and software libraries not included in the operating system.

All components have been configured and tuned to perform efficiently, scale linearly with the dimension of the repository and users, permit modular integration of new features, and be portable with respect to operating systems and middleware. We have used Fedora Linux distribution. The choice is motivated by the fact it is free, distributed with its source, and its licence is GPL (General Public Licence). GPL permits free distribution of software developed with its tools. This kind of software licence has been created by Richard Stallman, under many aspects the father of free software movement. The licence specify a legal state that is a sort of negation of copyright: the software subjected to GPL licence has to be free, distributed with its source code, and it has to be redistributable without limitations. Moreover, software derived by GPL software is itself subject to the same licence. In this way, the developed software cannot be sold, but it can only be freely distributed, which implies that enhanced versions of a free software are free too.

Fedora Linux has provided basic infrastructure of what is often called a LAMP (Linux/Apache/MySQL/PHP) architecture. Apache (http://www.apache.org) is the HTTP (Hyper Text Transfer Protocol) daemon, which is the program that manages on the server the web pages distribution. This program answers to page requests from a client. When a request arrives, it loads the page from the storage unit and sends it to the client, so that it can be displayed in the web browser. When the content of a website is dynamic, that means that the pages are dynamically built upon client requests, HTTP daemon needs to interface with the relational database management system. A program is needed to load the data from the database and to export them in HTML (Hyper Text Markup Language), so that the client web browser can visualize them. We decided to use PHP (http://www.php.org), which is an interpreted high level programming language that can be used to add dynamicity to web pages. The advantage is that PHP enables to use procedural and object oriented programming paradigms and it interpreters the syntax of many programming and scripting languages, such as C and C++ languages, and the scripting language of Unix System V shell. When a client requires pages that contains PHP code, those are interpreted and the instructions are executed by the server. In this way it is possible to incorporate in a page contents coming from a database. We use MySQL (http://www.mysql.org) as relational database managing system (RDBMS). It provides a software tool that is programmable following the specification of SQL (Structured Query Language) standard with performances that are comparable with commercial systems. Furthermore, it is well documented in many languages and has a large user base, which provide a large quantity of tutorials and technical articles.

In this same layer we find FTP (File Transfer Protocol) daemon and Java Servlets, The first provides basic file transfer functionalities. Those are available to SIRMM users for update operations. Java Servlets provide client server communication in all those features that are executed on client, but require server interaction.

Middleware offers components for secure communication, authentication, Java virtual machine and software libraries for remote access. Secure data transfer is obtained through a Secure Socket Layer (SSL) tunnel over FTP. We use OpenSSL server (http://www.openssl.org), that is the outcome of a free software project with the aim to implement SSL version 2 and 3 and the libraries that permit its usage within other software programs. Java applications are usually loaded with a HTML page and are executed within the client web browser with the Java Virtual Machine. JavaScript is an object oriented scripting language developed by Sun Microsystems and Netscape that can be inserted in HTML pages. Using JavaScript, it is possible to have access to client information, such as the number of visited sites, browser capabilities, the number and type of installed plug-ins and browser windows dimensions.

A library of API usable within JavaScript programming language and the applets support for Java have been used to provide a simple way to securely preview and upload all user data.

In the **Application general** layer, all management tools can be found that specifically operate on data, metadata and files. There is also a Web delivery component that uses JavaScript to handle differences in web clients used to access SIRMM. DB client component contains interfaces for DBMS administration. Those interfaces are used to alter tables adding, removing and renaming fields, to add database users with specific privileges. Data management component deals with the insertion, editing and removal of data in database tables. It assists the user to manipulate mathematical formulas, which are among data and need particular attention. Those are usually displayed on clients as images. This method has the drawback to cancel the semantic associated with the formulas. SIRMM treats them as MathML (http://www.w3.org/Math/) objects and stores them using Latex mathematical environment description language. MathML has the advantage to be the W3C standard markup language for mathematical formulas. It permits to maintain the semantic of formulas. In the future, all web browser will adhere to the standard and it will be possible to search for a particular formula, cut and paste formulas from one document to another. Latex is widely used by scientific community as markup language. It provides an environment to format mathematical formulas which is both complete and flexible.
File management component provides tools for file upload and download. SIRMM enables users to upload file on the server. This is useful to add a figure or a chart to contents. An authenticated user can upload a file using a secure channel. He can upload a file from its client, from a web site or from a third part FTP server. The user can preview the file before inserting it in the system.
Content management offers functionalities to manage learning content. It is used to manage, introduce or delete links among data stored in the database. The metadata management collects and distributes users information to various parts of the system. Users have different privileges and can perform different operations: when a user authenticates himself in the system, a server session is started, which maintains information about him. During the session, operations requested by the user will be completed only if he has the appropriate privileges. The session is closed either at user logout, or due to a connection error, for example when the browser window is terminated from the client side, or after a fixed time.
Web delivery deals with HTML display. Among its functionalities, it provides information about the client capabilities. For example, if the client browser can display formulas in MathML, it sends the formulas in that format, otherwise images containing the formulas are generated and sent.

In the **Application specific** layer we find the Application authoring component, which contains the design templates common to all parts of the system. They have been implemented in HTLM using CSS (Cascade Style Sheets). The pros of their use is that the content is separated by its representation, transfer time is reduced and it is possible to create different layouts for different devices (printers, textual browsers,…) with a single HTML page. Consumer and Producer management components comprise functions used by those who consume and produce information stored in the repository. They use the functionalities of the Application general layer in order to provide the web graphical user interfaces needed by users for their interaction with the system. Producer management deals with forms for data upload and editing. This components has an enhanced RTF (Rich Text Format) editor to insert text, images and formulas. The editor previews the pages before actual insertion in the system.
Content searching, at present not yet implemented, will enable semantic searches in the repository. System management gathers all procedures for back-ups, security, systems operation check, and resource monitoring. Application delivery focuses on user needs, such as the availability to the user of a specific web browser, and it will take care of specific needs, such as the use of specialized devices for content fruition.
The software architecture provides software packages in which users can interact without any concern about the underlying fabric and software infrastructure. Tools needed have been provided by the open source community and all software has been developed with the idea that its use into SIRMM is motivated by its ease of distribution, installation and use by the community. The system will be publicly available and will cross the frontiers of universities and other institutions, to create a virtual organization in which it is natural to share knowledge.

4. Using the system

SIRMM includes three types of users, with different roles, and therefore authorization levels: *consumer/student*, *producer/teacher* and *administrator*. The *consumer* is a user that navigates the system, he can read all records/forms of the database, search information and download files; he is in general a student that uses SIRMM as an interactive learning system, and he does not necessarily need to be authenticated by system. The *producer* is a user that can add information and data in the SIRMM database; usually he is a teacher who wants to insert and share Learning Objects (LO) and therefore needs to be authenticated from the system. Obviously a producer can modify only the LO that he has previously inserted. In addition, using the data of SIRMM the producer can build his own e-learning course. Finally, the *administrator* is in charge of all system management tasks, such as the authentication requests.
This structure has be implemented into SIRMM through three main sections: *Navigate*, *Contribute* and *Course*; any LO in SIRMM system can be accessed by the consumer (*Navigate* section), inserted and modified by the producer

(*Contribute* Section). He can also produce an e-learning course, through a logical path which includes LO belonging either to SIRMM or to the external world (*Course* section). Figure 5 shows the flow of information and activities into SIRMM, with the roles of the different kind of users.

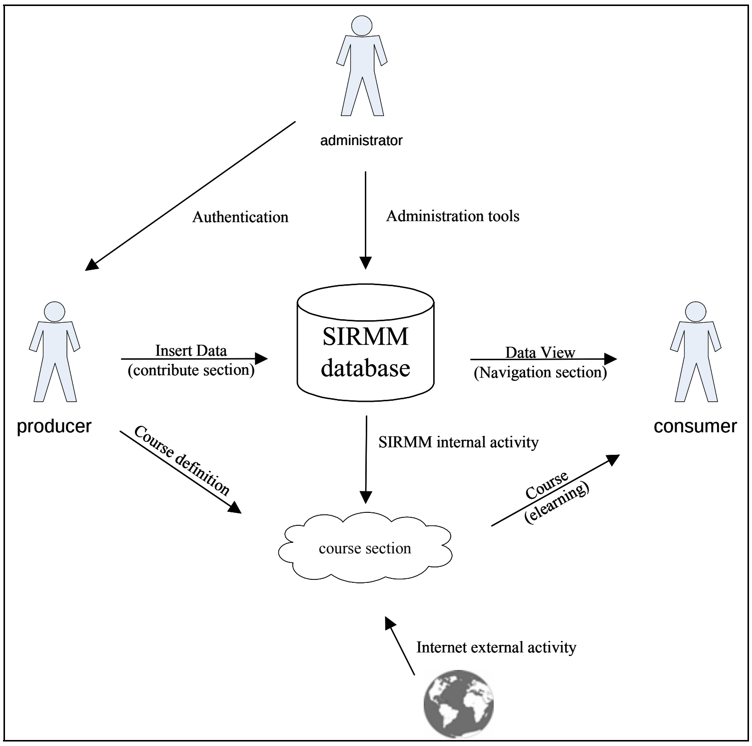

Fig. 5. SIRMM using interfacing structure.

The web interfaces provided by SIRMM have a rather simple full-screen menu structure (Figure 6), that includes:

- The *Horizontal menu* at the top of the screen which offers access to the functions that allow to use the system: starting from here, the user can either *Browse* through the system by accessing to information repository, or to *Contribute* with a new record or new *Courses*. In addition, the *Download* of external resources is allowed.
- The *Side menu* enhances the system navigation. It works dynamically and, at each moment shows only the entities that are related to the record of database that is currently displayed. For instance, if the user is accessing the information about a specific *Model*, then only *Problem*s, *Resource*s and *Software* links.
- The *Page content,* i.e. the area of the graphical user interface in which records are displayed.
- *Bottom bar* contains links to the Site Map and to the Contact us section. If the user has logged in as an authorized producer, it will display links to change user settings and to logout.

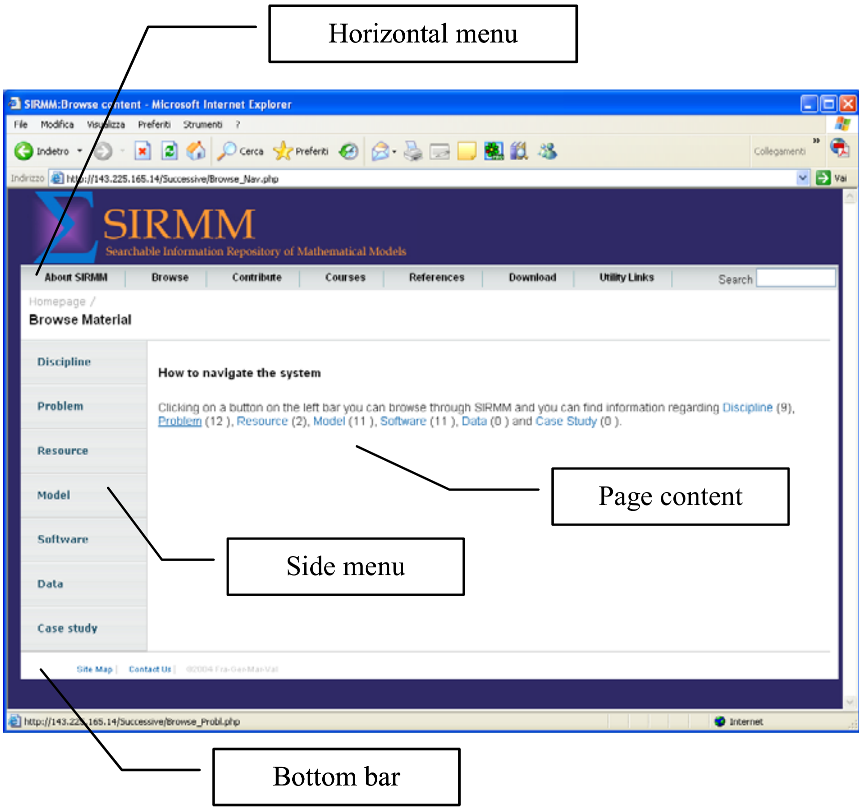

Fig. 6. SIRMM user interface.

3 Navigate Section

The navigate section of the system allows to browse all elements of the SIRMM data base. It is possible to navigate the system using the side menu (`discipline, problem, resource, model, software, data, case study`). When a specific instance is chosen the names of all related instances are shown, and the user can access the information of each element. Figure 7 shows a software instance (SIMILE, for system dynamics problems), in which it is possible to see the hypermedia web page with text, images, mathematical formulas. Moreover the side menu links directly to LO related to such instance, so to enhance possible interactive educational paths to the users.

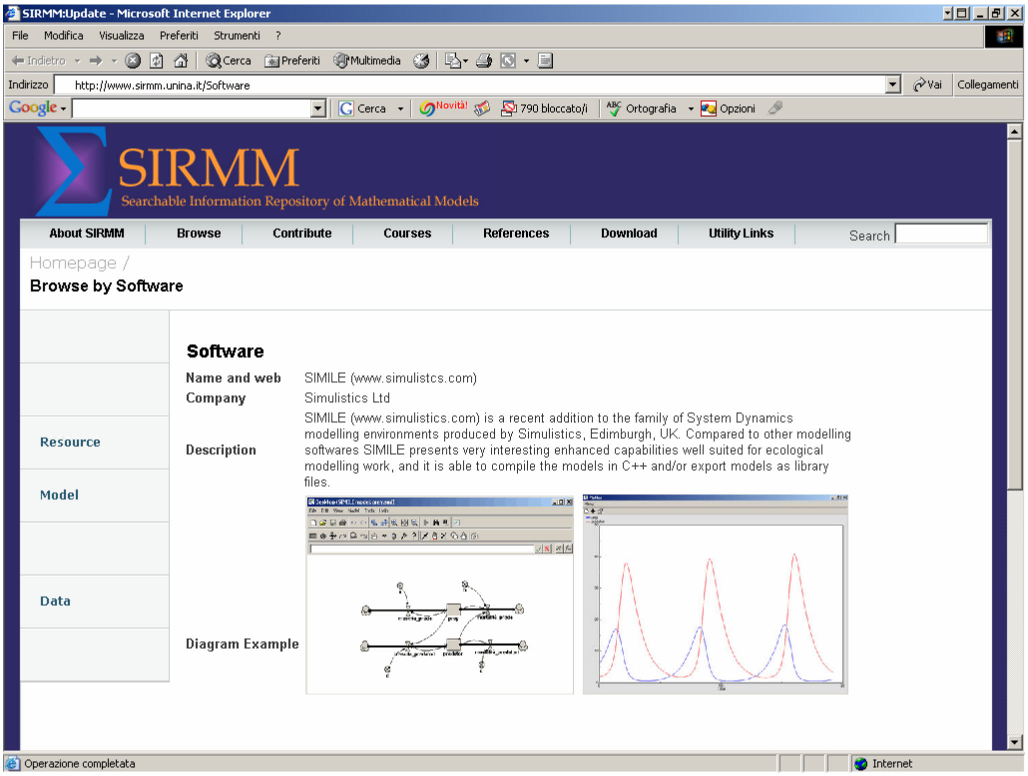

Fig. 7. SIRMM software instance.

4 Contribute section

The Contribute section of the system provides forms to insert or update elements in the system database. Once the user has been authenticated, he can add records to any entity in the database. He can use an improved version of the rich text format editor of the Wikipedia project (http://www.wikipedia.org/) to edit text, images and formulas. Formulas are inserted in the Latex mathematical environment syntax. They are then stored in the database and displayed either as gif images, or as MathML formulas, depending on the browser functionality. Moreover it is also implemented a *formula preview* window (Figure 8).

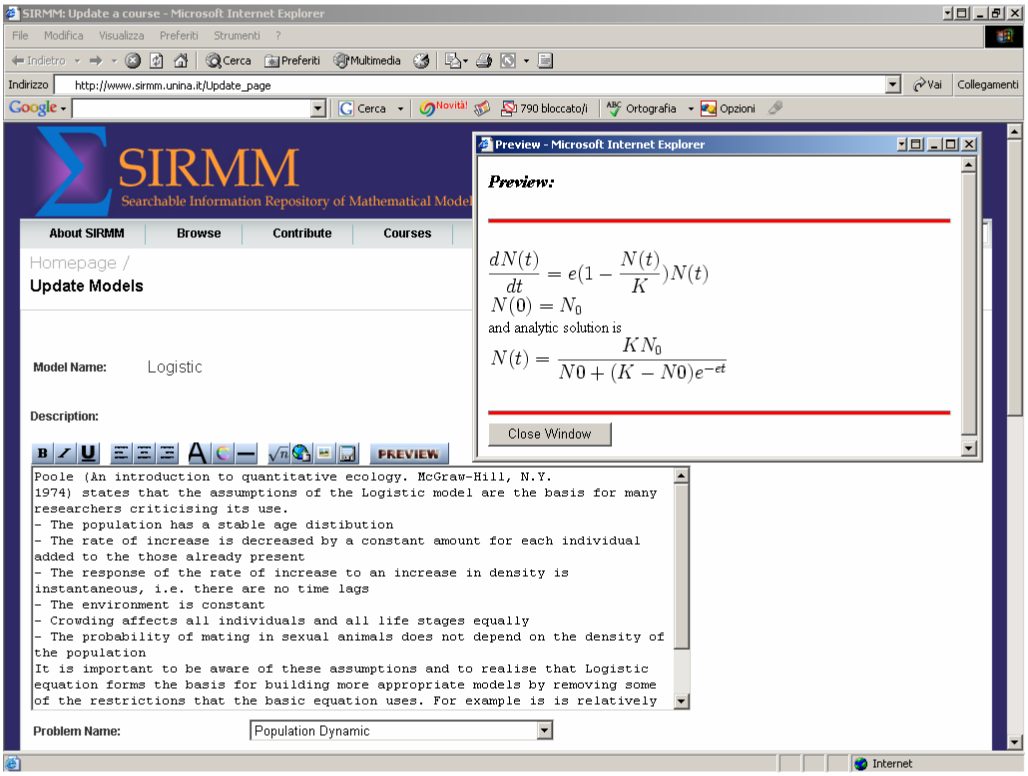

Fig. 8. Update page of Contribute section.

5 Course section

The *Course section* allows the consumer to access a course which has been already implemented in the SIRMM framework, while the producer can either modify his own course, or build a new course, possibly a modification of an already existing e-learning course. A SIRMM course is composed of two parts: a web page of information (name, teacher, schedule, credits, faculty, …) and a list of activities. Any activity is a link to a web page: an internal link to LO of the SIRMM database or an external link to other www pages.

5. The design of a SIRMM e-learning course

In this section we show by means of an example an application of SIRMM in designing an e-learning course, namely an already existing course of applied ecology. Rebuilding existing materials for an e-learning system does not simply mean moving traditional written training materials to the computer screen. Rather, the effectiveness of the training program we want to build strongly relies on the effectiveness of the strategies we follow to create and to convert existing course materials for online delivery and organizing them into an e-learning format. With respect to such purpose, SIRMM can be considered as an environment which helps to easily incorporating multimedia and interactive elements into an applied mathematics program, from the creation of an E-learning Course Syllabus, to the addition of contents in the form of lecture notes, handouts, assignments, and links to websites, where contents might belong or not to the SIRMM system.

The course of Applied Ecology was originally offered to undergraduate engineering students in the 2nd study semester. It is originally a lecture course based on a textbook and on commercially available software (Simulink, Matlab, etc.) and supplemented with weekly self-assessment exercises which are offered and corrected via the internet. The course aims to teach participants to develop simple models based on proper mathematical equations for the description and simulation of natural processes, with the main focus on the environmental and natural systems modeling. The course includes two modules: *population dynamic* and *ecological cycles*. In particular the population dynamic subject is easily introduced from simple models to complex models. The main goal of the course is to familiarize engineers with a number of mathematical and computational concepts to be used as modeling tool for ecological systems. The logical steps to create the course are the following ones:

1. Creating the e-learning Course Syllabus
2. Designing the e-learning Course into the SIRMM-course environment
3. Adding contents through the SIRMM system
4. Mapping the contents into the SIRMM-course environment

Steps 1-2 requires a clear overview of the study process which must be taken according to a well defined course flow, the course topics and their relationship to each other, the teaching strategy to follow and the material to be used, the information resources, the expected outcomes. The table 1 shows the Applied Ecology Course *Syllabus* must be translate in e-learning environment.

Table 1
Applied Ecology Course *Syllabus*

#	List of topics
1	Ecology principia
2	System dynamic modeling principia
3	Notes on the ordinary differential equation for system dynamic modeling
4	Introduction to system dynamic software: Stella, Simile, Simulink
5	Using Simile for modeling
6	Introduction to population dynamic
7	Isolate population dynamics: exponential model and logistic model
8	Interaction population dynamics: prey-predator model, competition model, cooperation model
9	The cycle in ecology
10	Model of ecological cycle: nutrient cycle, organic cycle, carbon cycle
11	Global change problem
12	Example of Global change model.

Figure 9 shows how Applied Ecology is a multidisciplinary course in which topics from mathematics, computer science and biology are combined in a common framework focusing on the basic population modeling issues.

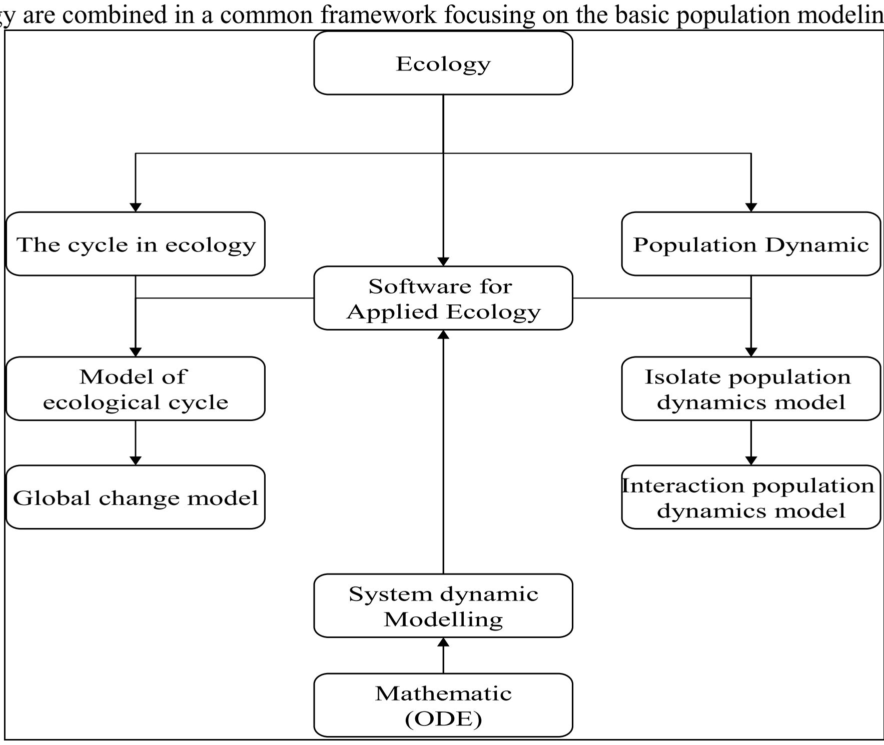

Fig. 9. Applied Ecology Course Flow chart.

Step 3 will require to create and convert existing course materials for online delivery and organizing them into a general and homogeneous e-learning format; as was already pointed out, in that respect the course section in SIRMM wishes to be a flexible and easy-to-use tool. In our case study, the “knowledge elements”, in the SIRMM’s data base are synthetically reported in Table 2. The producer (i.e. the teacher), using the login and password authentication procedure, inserts data and instances: the DISCIPLINE ecology (subject of the course), the two PROBLEMS (modules of the course) dynamic population and ecological cycles, some MODELS for each problem, and SOFTWARE description of a Simile system dynamic software. Moreover the teacher inserts all model files implemented in Simile, that can be downloaded by students and some DATA and RESOURCE to produce some case study.

Table 2: SIRMM elements for Applied Ecology course

#	Object type	Object Name
1	Discipline	Ecology
2	Problem	Population dynamic
3	Model	Malthus or exponential model
4	Model	Logistic model
5	Model	Prey – predator model
6	Software	SIMILE
7	Model	Competion model
8	Resource	Hannon, Ruth. Dynamic Modeling. Springer-Verlag
9	Data	data for Prey – predator model
10	Problem	ecological cycle
11	Model	Global change
12	Model	Nutrient cycle
13	Model	Organic cycle
14	Model	Carbon cycle
15	Data	Data for Global change cycle

Once all the elements/pages have been inserted, the producer is able to build and define his course through a system of links to the activity list (figure 10). These links can be sourced from SIRMM data base elements but also from "external" web resource. For the *Applied Ecology* course the first activity proposed by the teacher is a very general definition of "Ecology", using the predefined *SIRMM discipline form*, linked to two different *Problem* items (i.e. "Population Dynamic" and "Ecological Cycles"). Each problem has several *Models* and *Data*. For instance, "Population Dynamic" includes the classical one-specie models (exponential, logistic) and the two-specie models (prey-predator, competition, cooperation).

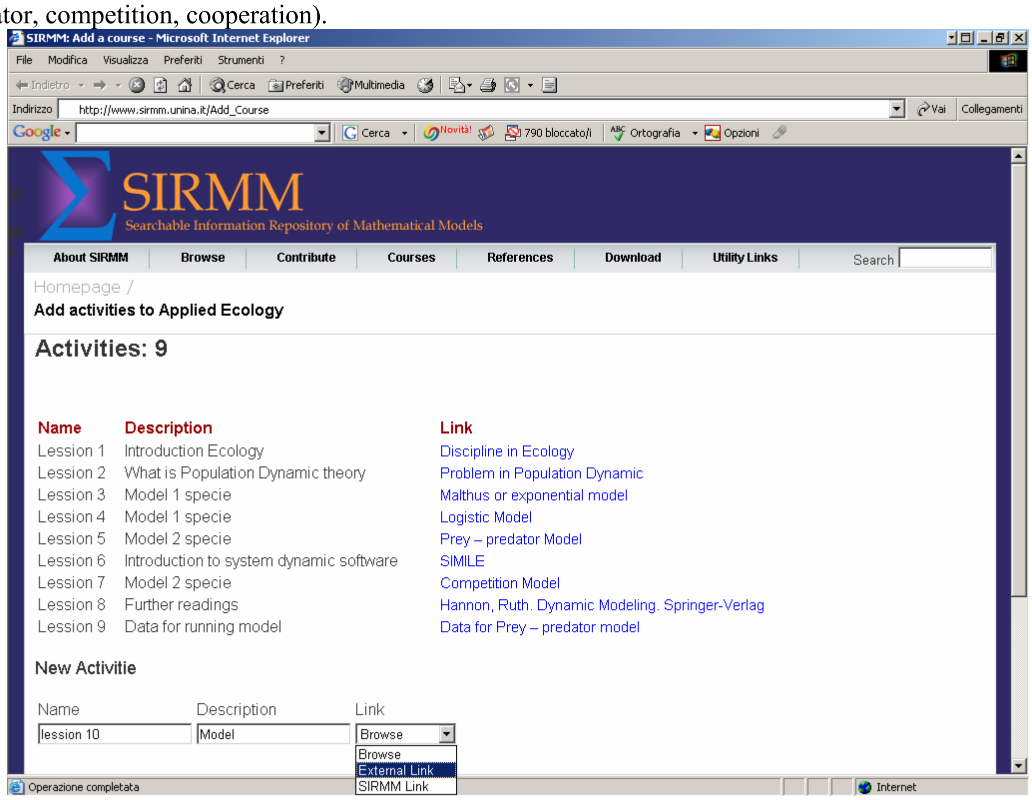

Fig. 10. SIRMM activity list of Applied Ecology course.

In order to exploit the availability of already existing material, the SIRMM course application allows to include any kind of external elements, such as resources available on the word wide web or any material available in electronic format, or simple links to interesting references. For the *Applied Ecology* course, the teacher inserts in the activity list some links to specific dynamic system software (e.g. software Populus 5.3 http://www.cbs.umn.edu/populus/) and some external links for students further readings (Darwing's specie definition, data for some prey – predator case study, concept of ordinary differential equation model, numeric aspects for models resolution). Figure 11 shows a proposed activity list for Applied Ecology course and the learning path suggested to students with internal and external element links.

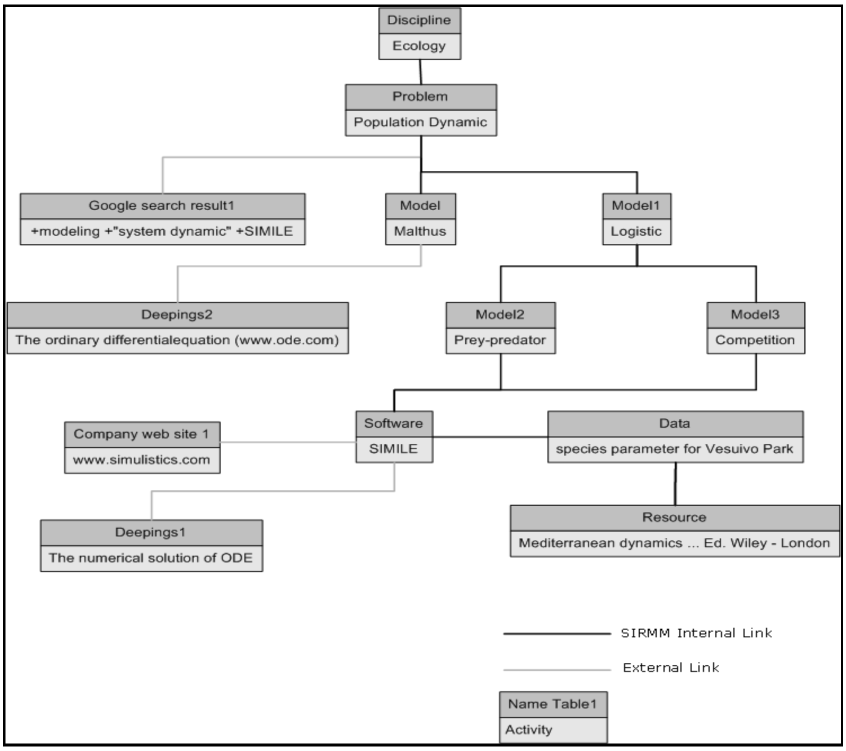

Fig. 11. Activity list for Applied Ecology course with internal and external element links.

Modularity and Flexibility of SIRMM structure support the creation of independent and well differentiated learning paths based on learning objects (LO) belonging to the system. Moreover, flexibility leads to high level of reusability of the in SIRMM, as well as of entire learning path (courses) that can be easily modified.

A way to exploit the modularity of SIRMM, is to implement a course by using learning units used in another course; completely different educational paths can be mapped on large sets of common forms and data. For instance, Figure 12 shows the e-learning paths of two different course units with large overlap on the sets of LO they are based on: Applied Ecology (dash line) and Macroeconomy System (solide line). In this example the course of Applied Ecology is defined by a procedural approach moving from the definition of discipline, and arriving through a model to sample data, whilst the Macroeconomy course follows an alternative path from data to solution and definition of the problem.

A different view of modularity is to offer the possibility to implement different paths through the material matching the instructional approach to the needs and interests of every student; this is what is called “differentiated instruction” (http://www.glencoe.com/sec/teachingtoday/subject/di_meeting.phtml), an instructional theory that allows teachers to take diverse student factors into account when planning and delivering instruction. Based on this theory, teachers can structure learning environments that address the variety of learning styles, interests, and abilities found within a classroom. Differentiating instruction is actually an essential tool for integrating technology into classroom activities in an effective way.

Of course, the reusability, is a feature that depends on the size and on the quality of the learning objects. Wiley, Recker, & Gibbons, (2000) noted the inverse relationship between the size of a learning object and its re-usability. As the learning object’s size decreases (lower granularity) its potential for reuse in multiple applications increases. Therefore, the potentiality of SIRMM strongly depends on how learning objects will be supplied to the system, i.e. on the way the suppliers of the information (users) will feed the system. It is planned, in the next future, to offer templates to foster users to produce their learning objects as small granularity objects, possibly integrated with already existing objects so to make easier the assembling of learning objects (content packaging).

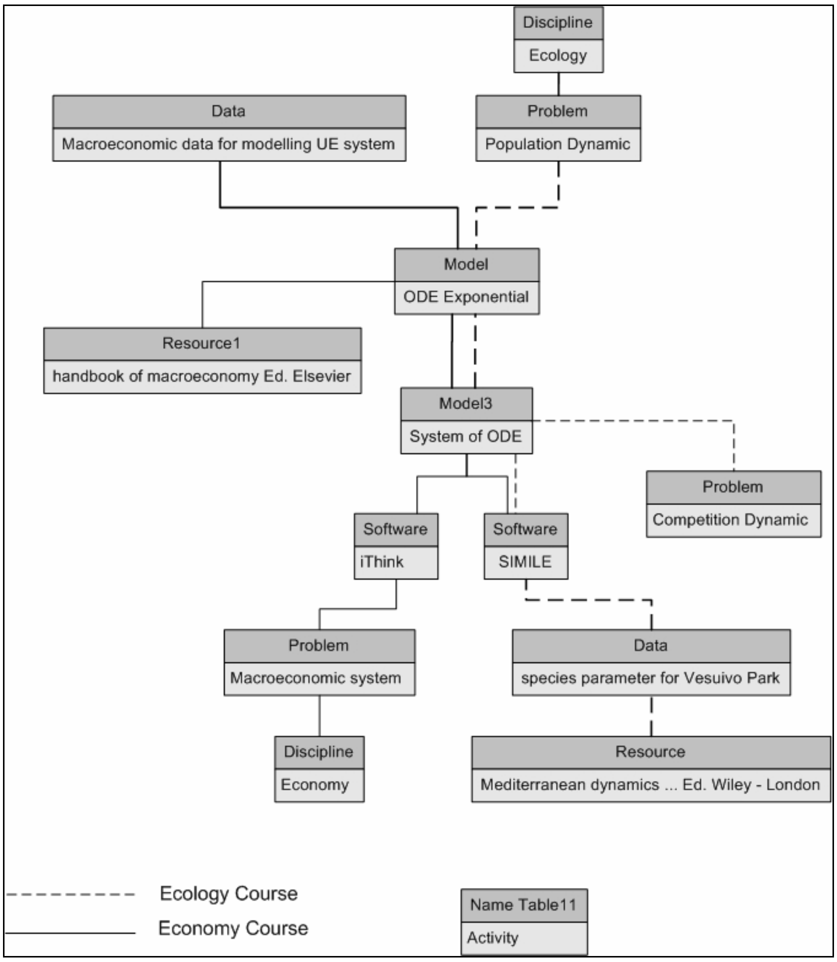

Fig. 12. The "structure intersection" of two SIRMM course.

6. Conclusions and Future Work

We believe the use of a web tool in a cognitive process provides a powerful device, especially if it mimics the learning process used in everyday life. In our experience, knowledge is based on information that is acquired by different resources, which are searched when needed, and that is processed to give a meaningful answer to our problems. Resources act in a peer to peer fashion, providing and acquiring the information needed, thus realizing a constructive process. In the very same way, a user of a learning environment should be able to search for more information from other resources when it is needed, and to process it in a meaningful way. Interconnecting web tools so they can share their contents makes it possible to obtain distributed learning system in which knowledge is shared. A viable technological solution to the interconnection problem is a peer to peer model, similar to the one that is used in many well known system for file sharing.

This approach has also some implications. As we experience in everyday life, the way in which we name objects and concepts varies; moreover there is no single way to describe processes that produce knowledge. This means a distributed and interconnected system is needed to match different terminologies and models in which a concept can be expressed, so that knowledge can be shared and reused. A way to specify a conceptualizations are ontologies (Gruber, 1993). We may choose to write an ontology as a set of definitions of a formal vocabulary. Although this isn't the only way to specify a conceptualization, it has the property that semantics is independent of consumer and context. If we agree to use a vocabulary in a way that is consistent with respect to the theory specified by an ontology, then we can build systems that commit to ontologies and we can share knowledge with and among these systems. Of course, there has to be an agreement on the ontology to use in a specific field, otherwise it will be difficult to make them interact. This is an ongoing and very promising research topic, which is producing tools for semantic aware web services that are well suited for our purposes.

Acknowledgements

This work has been partially founded by the Italian MIUR (Ministero dell'Istruzione, dell'Università e della Ricerca) within research projects "Distant learning and knowledge integration in life sciences", "New technologies of control, treatment, and maintenance for the solution of water emergency", AQUATEC (under PON 2003-2005 contract), and the M.I.U.R. FIRB project "Large Scale Nonlinear Optimization" no. RBNE01WBBB.

References

Avgeriou, P., Papasalouros, A., & Retalis, S. (2001). Learning Technology Systems: issues, trends, challenges. *Proc. of the 1st Int. Organization for Science and Technology Education Symposium in Southern Europe: Preparing future citizens*.

Avgeriou, P., Retalis, S., & Skordalakis, M. (2001). An Architecture for Open Learning Management Systems. *Panhellenic Conference on Informatics*, 183-200.

De Corte, E., Verschaffel, L., & Greer, B. (2000). Connecting mathematics problem solving to the real world. *Proceedings of the International Conference on Mathematics Education into the 21st Century: Mathematics for Living*, 66-73.

Dede, C. (1996). The evolution of distance education: Emerging technologies and distributed learning. *The American Journal of Distance Education*, 10(2), 4-36.

Dick., W. (1992). An instructional designer's view of constructivism. In T. Duffy & D. Jonassen (Eds.), *Constructivism and the technology of instruction: A conversation*. Hillsdale.

Eastmond, D., & Ziegahn, L. (1996). Instructional design for the online classroom. In Z. L. Berge, & M. P. Collins (Eds.), *Computer mediated communication and the online classroom*, 59-80. Cresskill, NJ: Hampton Press, Inc.

Gallopoulos, E., Houstis, E.N., & Rice, J.R. (1994). Computer as Thinker/Doer: Problem-Solving Environments for Computational Science. *IEEE Computational Science & Eng*., 1(2), 11-23.

Gallopoulos, E., & Sameh, A. (1997). CSE: Content and Product. *IEEE Computational Science & Eng.*, 4, (2), April-June, 39-43.

Giannino, F., Guarracino, M.R., Monetti, V.M., Romano, N., & Toraldo, G. (2004a). SIRMM: Searchable Information Repository of Mathematical Models. *EISTA 2004*, *International Conference on Education and Information Systems: Technologies and Applications*, Orlando (Florida, USA).

Giannino, F., Guarracino, M.R., Monetti, V.M., Romano, N., & Toraldo, G. (2004b). An Adaptable Learning Technology System for Mathematical Models. *ICALT 2004, The 4th IEEE International Conference of Advanced Learning Technologies*, Joensuu (Finland).

Greer, B. (1993). The modelling perspective on the wor(l)d problems. *Journal of Mathematical Behavior*, 12, 239-250.

Greer, B. (1997). Modelling reality in mathematics classroom: the case of word problems. *Learning and instruction*, 7, 293-307.

Gruber, T. R. (1993). A translation approach to portable ontologies. *Knowledge Acquisition*, 5(2), 199-220.

Hannafin, M., Land, S. & Oliver, K. (1999). Open learning environments: Foundations, methods, and models. In C. M. Reigeluth (Ed.), *Instructional design theories and models: A new paradigm of instructional theory (Vol. II)*, New Jersey: Lawrence Erlbaum Associates, 115-142.

Harasim, L. (1996). Online education: The future. In T. M. Harrison, & T. Stephen (Eds.), *Computer networking and scholarly communication in the twenty-first-century university* (pp. 203-214). New York: State University of New York Press.

Houstis, E.N (2003). The Role of Problem Solving Environments in Engineering and Mathematics Education. *Proceedings of the 6th International Conference on "Technology in Mathematics Teaching". Telematics Network Centre, University of Thessaly*.

Jonassen, D.H., & Grabowski, B.L. (1993). Handbook of individual differences, learning and instruction. Hillsdale, NJ: Lawrence Erlbaum Associates.

Kong, S. C., & Kwok, L.F., (1998). An interactive teaching and learning environment for graph sketching. *Computers & Education*, 32, 1-17.

Landau, R. (1995). Computation in Modern Physics. *IEEE Computational Science and Engineering*, 2(3), 84-85.

Lowe, D., & Hall, W. (1999). Hypermedia & the Web: An Engineering Approach. *John Wiley Ltd.*.

Marchioro II, T., Martin, D. & Payne, W. (1995). UCES: An Undergraduate CSE Initiative. *IEEE Computational Science and Engineering*, 2(3), 69-73.

Marchioro II, T. L., & Landon, R. H. (1997). Web-Based Education in Computational Science and Engineering. *IEEE Computational Science & Eng.*, 4, April-June, 19-25.

McCormack, C., & Jones, J.D. (1997). Building a Web-based Education System. *Wiley Computer Publishing*.

Michailidou, A., & Economides, A. (2003). Elearn: Towards a Collaborative Educational Virtual Environment. *Journal of Information Technology Education*, 2, 131-152.

Moallem, M. (2001). Applying Constructivist and Objectivist Learning Theories in the Design of a Web-Based Course: Implications for Practice. *Educational Technology & Society*, 4(3).

Oppermann, R., Rashev, R., & Kinshuk, (1997). Adaptability and Adaptivity in Learning Systems. *Knowledge Transfer VII*, Ed. A. Behrooz, pAce, London, 173-179.

Poggio, T., & Smale, S. (2003). The Mathematics of Learning: Dealing with Data. *Notices of the American Mathematical Society (AMS)*, 50(5), 537-544.

Pohjolainen, S., Hantakanges, S., Ranta, P., Levasma, J., & Pesonen, K. (2003). A learning experiment in mathematics using A&O-learning environment. *Int. J. Cont. Engineering Education and Lifelong Learning*,13, Nos. ½.

Quarteroni, A. & Saleri, F. (2003). Scientific Computing with MATLAB. Heidelberg: *Springer-Verlag*.

Rafaeli, S., Barak, M., Dan-Gur, Y., & Toch, E. (2004). QSIA – a Web-based environment for learning, assessing and knowledge sharing in communities. *Computers & Education*, 43, 273-289.

Sheremetov, L., & Arenas, A.G. (2002). EVA: an interactive Web-based collaborative learning environment. *Computers & Education*, 39, 161-182.

Smith, P. L., & Ragan, T. J. (1993). *Instructional design.* Upper Saddle River, NJ: Prentice-Hall.

Vrasidas, C. (2000). Constructivism versus objectivism: Implications for interaction, course design, and evaluation in distance education. *International Journal of Educational Telecommunications*, 6(4), 339-362.

Wiley, D. A., Recker, M., & Gibbons, A. (2000). Getting axiomatic about learning objects. Retrieved from the Internet on May 5, 2004 from http://reusability.org/axiomatic.pdf