Design and Development of Computer-Aided Chemical Systems: Virtual Labs for Teaching Chemical Experiments in Undergraduate and Graduate Courses

I. Luque Ruiz,* E. López Espinosa, G. Cerruela García, and M. A. Gómez-Nieto

Department of Computing and Numerical Analysis, University of Córdoba, Campus Universitario de Rabanales, Edificio C2, Planta-3, E-14071 Córdoba, Spain

Received February 28, 2001

An environment for the construction of virtual chemistry experiments is presented. This environment is based on the $E(V) = M + m$ model—***E****xperiment (****V****irtual)* = ***M****aterials* + ***m****ethod*—proposed and described herein, which allows the representation and subsequent building of chemistry experiments in virtual 3D worlds to any degree of complexity. The object-based nature of the environment not only allows its use on the Internet but also facilitates integration with other systems, while enabling the system to represent and organize knowledge in such a way that it is available to any teaching environment dealing with chemical laboratory experiments.

1. INTRODUCTION

While problems arising in the teaching of chemistry at high school, graduate, and undergraduate level may not be vastly different from those of other experimental disciplines, they are still specific to the practical teaching of chemistry and require a directed approach, as described in other studies. The need exists for an adaptation of methods and teaching media to the new computer technologies since students are well-aware of and influenced by the latest developments in computer technology at a practical or sociocultural level and will also be required to work in computer-oriented environments.[1−6]

The efforts of the chemical community to adopt these new technologies may be clearly seen in the great number of studies published, particularly in the past decade, and in the vast amount of information, which is today available on the Internet.[7−18] Public and private organizations, high schools, universities, and professionals in the private sector have been producing—for general or restricted access—an ever-increasing stream of teaching material of varying degrees of complexity for the use of students and professionals in the world of chemistry teaching.

This published material may be grouped as follows:

Tutorials: teaching material dealing with specific aspects of chemistry and found as hypertext documents (html) or formatted for one of the standard word processors.[20,21]

Support software: small-to-mid-sized software programs aimed at both teaching and enabling the student to solve chemical problems, or storage, retrieval and processing of chemical information. This software is mainly directed at areas such as periodic table information, balancing reactions, detailed composition, calculation of molecular masses, viewing molecular structures, interpretation of spectra, etc. In many cases, the software is accompanied by tutorials that "explain" the basics and the chemical, mathematical, and deductive processes to follow in order to solve the given problem.[21−26]

In this group and closely related to the foregoing we have intelligent tutorial systems (ITS), including such categories as simulations (the prevailing category), learning systems and knowledge-based systems.[27−30]

Laboratory software: software designed to simulate teaching lab experiments. Such products have recently come much to the fore, with teaching and research centers investing great effort in development owing to reasons discussed in the next section of this paper. Applications in this category would include the following: *Chemlab*, an excellent computer-based lab simulator that includes a wide range of experiments such as acid−base titration, precipitation, manipulation of materials, gravimetric analysis, calorimetry, and reaction kinetics and comes with utilities that would otherwise be classed within the other aforementioned groups, *Electro-Chemical Cells Pro*, which allows the performance of redox experiments in a 2D environment, and the *IrYdium Project*, software developed by the University of Massachusetts, which comprises a 2D virtual lab developed in Java and where lab experiments can be designed and carried out. Further commercial programs are listed in the literature.[31−33]

As interest in the development of virtual labs soars, there arises a series of issues with respect to chemistry teaching and found in a great number of recent programs, namely the following:

1. They are developed for a fixed platform or operating system, which reduces the number of possible users.

2. Some require local installation for their use and will only run on that local model, while others may be run solely on the World Wide Web.

3. In some cases one (or a small set) of the experiments is predefined, together with parameters, materials, and all the physicochemical items required to run it, leaving the user unable to modify, much less expand, the experiment.

4. The user interface does not present the user with a virtual lab environment; at best there is a classical 2D

* Corresponding author phone: +34-957-212082; fax: +34-957-218630; e-mail: ma1lurui@uco.es.

10.1021/ci010015w CCC: $20.00
Published on Web 07/03/2001

interface, although some systems do provide a 3D view of molecular structures.

5. Only occasionally do these systems present a global approach to the teaching of experiments, providing the user with teaching material (tutorials), exercises, and observations, alongside experimental procedures and a dialogue system to assist in learning.

Furthermore, the bulk of this developed material is either privately held or cannot be reused by other systems due to being written in different languages or operational modes and sometimes to different standards, all of which means that educators are forced to expend effort in the redundant construction of teaching materials using the new technologies.

The present study describes the $E(V) = M + m$ model—***E**xperiment (**V**irtual) = **M**aterials + **m**ethod)*—for the building of chemistry lab experiments in a virtual environment. The model is suitable for use in the construction of an ITS, as it offers the following features: (1) inclusion of the teaching component, since content may be ordered and sequentialized for adaptation to both the student and the experiment; (2) simple representation of the domain of the problem (the experiment), specialized in blocks ordered according to the teaching process; (3) easy integration of other modules forming part of the ITS, such as that of the student, as well as the integration of other refinements; and (4) a communication interface created in line with current standards (based in windows interface), which facilitates user-system interaction.

This paper first describes the objectives of the study and then in section 3 provides a description of the proposed paradigm. The description includes specification of the structure and content of experiments in an object-oriented model, allowing both the classification of knowledge in learning blocks and its sequencing in the teaching process. Section 4 describes the operating prototype and its static operational architecture, closing with a discussion of the work that mentions future projects and enhancements currently under development.

2. OBJECTIVES AND BASIS OF THE MODEL

The present project aims to tackle certain of the problems reported in the teaching of chemistry lab experiments; as with other researchers, the present authors believe that the new technologies may contribute toward a solution.[4,6−8,10,30−33]

Generally speaking, these teaching problems may be broken down as follows:

1. Cost: the high cost to educational centers incurred by practical classes, where substantial investment in instruments, apparatus, reagents, and so on may frequently result in fewer lab sessions, often determined by expenditure.

2. Safety: especially in the case of high school students, the risk involved in manipulating chemicals and certain instruments and materials leads to the avoidance of dangerous experiments and cuts down on practice time.

3. Time: the shortage of time allocated to practical work in current curricula means that very few experiments can be performed; equally, the student will have little or no chance of repeating the task for greater understanding and learning.

4. Space/Size: lab space and the size of student groups attending practice sessions, in addition to the abovementioned factors, will sometimes mean that not all students may be involved in the manipulation of materials involved in the experiment.

5. Lack of motivation: the 21st century student requires motivational techniques adapted to the present day. These should encourage the use of modern technology in the same way the student uses this technology to carry out everyday social activities. The teaching of practical chemistry through traditional lab sessions does not, however, greatly promote student participation, perhaps then calling for the introduction of modern technology.

All these issues present a complex situation for teaching centers relying on conventional techniques and procedures; the introduction of modern methods and procedures should, however, provide an answer for at least some of these problems, while increasing student motivation.[1,2,8−10]

In the foregoing section, reference was made to ongoing studies aimed at tackling these problems; great effort has been made to adapt the new technologies to the teaching of chemical knowledge—theory, concepts, techniques, and procedures.

Clearly then, many teachers and researchers believe that use of the computer, and particulary Internet, can motivate the student, while reducing attendance requirements and thus the cost, risks, and so on involved in the teaching of chemistry. However, the form in which such knowledge is placed at the disposal of the student and how it is used may have a direct effect on its success.

The present study is based on a solution that, making use of the latest technology, allows the interactive transference of knowledge, making the most of resources in an environment of shared experiments and results and should lead to greatly increased student motivation.

To achieve these objectives, the authors propose the following:

1. Chemistry practical work should be split between traditional, hands-on teaching (where possible), and computer-based virtual experiments performed on the Internet (again, where possible).

2. Teaching resources employed should be available in the public domain. In other words, both the required software resources and the theoretical and practical content should be available for use by anyone with Internet access.

3. The technological or virtual teaching content should permit the reuse of software applications and their content.

4. The online aspect of teaching should be enacted using virtual laboratories where any kind of chemistry experiment could be performed, from any location or computer.

5. Chemistry experiments should be able to be performed at varying levels of complexity, in line with different educational grades.

6. Experimental content should be reusable, allowing its partial or complete use in the building of new experiments.

7. The performance of such practical work should involve the technical training of the student in line with the complexity of carrying out the experiment.

Thus, virtual labs may be viewed as another type of ITS where a suitable ontological model is used to build system components based on "chunks" of knowledge.[28−30,34,35]

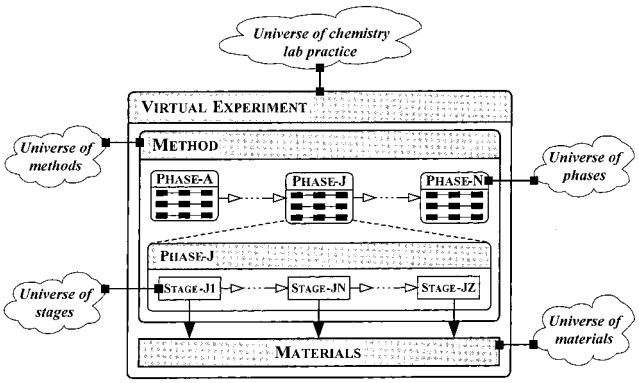

Figure 1. Context diagram of the $E(V) = M + m$ paradigm.

3. THE $E(V) = M + m$ PARADIGM

The paradigm $E(V) = M + m$—***E**xperiment(**V**irtual) = **M**aterials + **m**ethod*—which we propose in this paper is based on the concept that a virtual experiment may be described by a method which uses a set of materials.

The paradigm employed is based on the supposition that a virtual experiment may be described by a method which uses a set of materials.

Running a virtual experiment, performed in an interactive computer environment using a system of dialogues, the user may experience a 3D presentation simulating a chemistry experiment. Each experiment takes place in a virtual environment containing all the materials with which the user might interact—the virtual laboratory. Labs may be shared by multiple experiments and may also be built using different technologies and software, so user interaction will depend on the technology employed to build them.

Each virtual experiment has an objective and is thus accompanied by descriptive content aimed at delivering knowledge for the user to learn. While the nature and extent of the experimental content may be highly varied, presentation and user interaction is independent of the experiment.

This standardization of presentation and user interaction facilitates assimilation and leads to the user easily adapting to the virtual learning model.

Each experiment has a unique associated method. A method is a precise description of the tasks and resources used in the experiment.

Each method comprises structure and content. The structure of the method lists the processes to be carried out during the experiment, while the content includes all necessary information and the way in which this is manipulated therein.

The method's structure is made up of a sequence of phases. Each phase involves a global task or activity performed during the experiment, which may be broken down into further elementary tasks called stages.

The level of abstraction for the method's structure is independent of the method and thus of the experiment, thus allowing any experiment to be described at various levels of complexity.

This structural breakdown of an experiment, in which different knowledge blocks are organized chronologically in a working network, is a key factor in user-learning.[5,35]

Figure 1 shows a context diagram of the paradigm $E(V) = M + m$. It may be seen that a virtual experiment represents the knowledge of a laboratory experiment through a method structured in phases, which in turn are divided into stages where the materials are manipulated. These structural elements of experiments are extracted from a general universe, enabling them to be reused in the same or indeed in a different experiment.

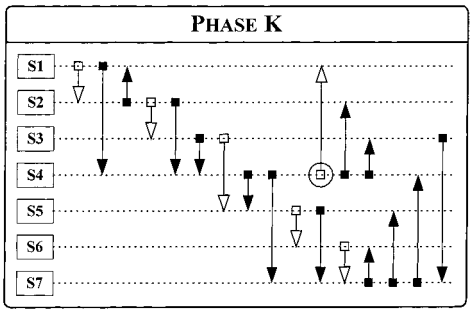

Figure 2. Diagram showing permitted transitions for phase-k.

3.1. Structure of the $E(V) = M + m$ Paradigm. Let us now examine the structural elements of the model using a bottom-up approach, according to complexity.

3.1.1. Stages. Stages are basic tasks in the structural description of an experiment. Depending on the level of detail required for the description of an experiment, a task may vary from a simple manipulation such as "Obtain a pipet" to one like "Obtain a pipet, fill it, and empty the contents into an Erlenmeyer flask".

Stages imply manual or automated processes carried out in the lab and involving some laboratory material.

They are distributed throughout the experiment, grouped in ***Phases***. Within a phase, stages comprise a sequence of tasks that may be described by one or more directed graphs.

Take a *phase-j* consisting of "Pour 10 mL HCl from the stock bottle into an Erlenmeyer flask". This phase may be defined by the following stages:

PJS1	Obtain a 10 mL pipet from the rack.
PJS2	Pipet 10 mL HCl from the bottles.
PJS3	Empty the pipet's contents into the Erlenmeyer flask.

Clearly the only possible order to carry out the stages from *phase-j* would here be ***PJS1***, ***PJS2***, ***PJS3***.

Suppose now that we wish to represent the same process in greater detail—*phase-k*:

PKS1	Obtain a 10 mL pipet from the rack.
PKS2	Open the bottle of concentrated HCl.
PKS3	Pipet 10 mL HCl from the bottle.
PKS4	Take the Erlenmeyer flask from the shelf and place on the bench.
PKS5	Empty the pipet's contents into the Erlenmeyer flask
PKS6	Place the pipet in the sink.
PKS7	Close the bottle of concentrated HCl.

Figure 2 contains a transition diagram[36] showing the different routes that could be taken to perform *phase-k* (for a clearer description, the user is assumed to be reasonably skilled). Again in Figure 2, stages are represented by labeled boxes: shaded boxes show stages that could be the start of *phase-k* (*S1*, *S2*, *S4*), and the arrows indicate the transitions allowed between stages. If when performing *phase-k* the user chose to begin at stage *S1*, the next permissible stage would be *S2* or *S4*. If the user opted to follow the transition to *S2*, the next stage could be either *S3* or *S4*.

If the transition to *S3* were chosen, the next move would have to be to *S4*, since getting to *S5* would suppose a transition prior to *S3*, that is, an Erlenmeyer flask must be present.

Alternatively, if the user had decided upon a transition to *S4*, the next move must be to *S3*, since moving on to *S5* would require a transition prior to stage *S3*, that is, the pipet has to be filled, and moving to *S7* would mean that not all the transitions to all the stages of *phase-k* could be completed by any of the existing routes.

Thus, if *phase-k* is started at *S1* it can be performed suitably or correctly by the system using any of the following routes:

R1:	S1, S2, S3, S4, S5, S6, S7	**R2**:	S1, S2, S3, S4, S5, S7, S6
R3:	S1, S2, S3, S4, S7, S5, S6	**R4**	S1, S2, S4, S3, S5, S6, S7
R5:	S1, S2, S4, S3, S5, S7, S6	**R6**:	S1, S2, S4, S3, S7, S5, S6
R7:	S1, S4, S2, S3, S5, S6, S7	**R8**:	S1, S4, S2, S3, S5, S7, S6
R9:	S1, S4, S2, S3, S7, S5, S6		

White arrows in Figure 2 show the permitted route that is considered, on defining *phase-k*, as the most recommendable and which will be taken into account by the system for the automatic execution of this phase without user intervention.

At each stage at least one ***Material*** is required for manipulation. The use of a material on occasion implies the presence of other materials, as shown by ***PKS3*** where a pipet is used—thus implying the existence of the bottle of concentrated HCl. Manipulation of material implies the following: (1) the presence of the material as well as those materials thereby affected and (2) a condition where the material manipulated and any other intervening material are in such a state that the manipulation may be successfully performed.

In the same foregoing example of ***PKS3*** the bottle of HCl must be open to allow the pipetting of 10 mL (a prior transition to ***PKS2*** has taken place) and the pipet has already been obtained (a prior transition to ***PKS1*** has occurred).

Thus, Figure 2 shows all the possible transitions that represent manipulation processes involving *phase-k*. However, from any given stage only some of these transitions will be allowed for the performance of *phase-k*, depending on the transitions previously carried out and therefore on the current state of the material.

The permitted routes for the user to perform a generic phase *n* will be those that (1) run through, with no repetition, all of the stages comprising *phase-n* and (2) for any given stage perform only those transitions that do not violate the priorities existing between the stages of *phase-n*, as defined by the state of the material.

3.1.2. Phases. Phases are global elements in the task network that detail a method and group together stages, amounting to a general task. Regarding an experiment, a phase is an activity that leads to an elementary step in the experiment and in which a recognizable, measurable result is produced.

Like stages, phases may be defined to varying degrees of abstraction and thus represent experimental activity at different levels of complexity. For example, an experiment to titrate 0.1 M NaOH with 0.1 N HCl may be either described as a single phase consisting of the titration process or said to comprise the following sequence of phases:

P1	Prepare a 0.1 M NaOH solution.
P2	Prepare a 0.1 N HCl solution.
P3	Titrate 0.1 M NaOH with 0.1 N HCl.

Again in the same way as stages, phases are laid out in a time-ordered route throughout a ***Method*** that describes a virtual experiment.

3.1.3. Methods. Methods describe the structure and chronological route of activities as occurring in the course of an ***Experiment***. A method is an abstract sequence of procedures representing the tasks to be carried out, the material to be used and its manipulation in order to successfully perform a chemistry experiment.

Since phases and stages may be described at various levels of abstraction, one laboratory experiment can be described by different methods at various levels of complexity, but as shall be seen in the next section (description of content), the user may carry out an experiment using a method at various levels of interaction according to the level of detail and complexity sought for the interaction.

3.1.4. Experiments. The term "experiments" is used to describe practical laboratory experiments whose operational technique and underlying knowledge are to be passed on to the student through a virtual environment.

In the $E(V) = M + m$ paradigm, an experiment is defined as the use of a method to manipulate a set of ***Materials***. Thus each experiment has an associated method that is described by the phases and stages in which the materials are employed.

Furthermore, an experiment has an associated virtual world containing the materials to be used and where the activities described in its associated method are carried out.

3.1.5. Materials. Materials represent the physical elements recognized in the virtual world or laboratory associated with an experiment and which can be used therein.

The materials category embraces a great variety of laboratory items, differentiated by their properties and associated functionalities.

Hence, the context of material includes such diverse items as chemicals (e.g. existing products), furniture (e.g. lab benches), material (e.g. glassware), or instruments (e.g. pH meters).

3.2. Content of the $E(V) = M + m$ Paradigm. Modeling of the $E(V) = M + m$ paradigm was performed using the object-oriented paradigm, and complies with the UML standard.[37,38] Using the object model, a system can be described by a set of objects listing the properties and behavior of the system and each of its component parts.

The class diagram in Figure 3 shows the main entities of the static model representing the $E(V) = M + m$ paradigm, as follows:

A *Component* is considered to be any object existing in the system and may in turn be formed by other components.

The *Component* class is an abstract generalization class. By the term "abstract" we mean that objects belonging to the generalization class must be refined to one of the specialization classes of which it is composed.[38] The dynamic keyword shown in Figure 3 is a UML stereotype. When a stereotype is applied to an element such as a node or a class, new elements related to the vocabulary of a domain are being introduced; these are like primitive building blocks. Each stereotype may provide its own set of labeled values, semantics, notation, and so on.

Here, the stereotype enables us to indicate that the generalizations may be updated for a given object. When a

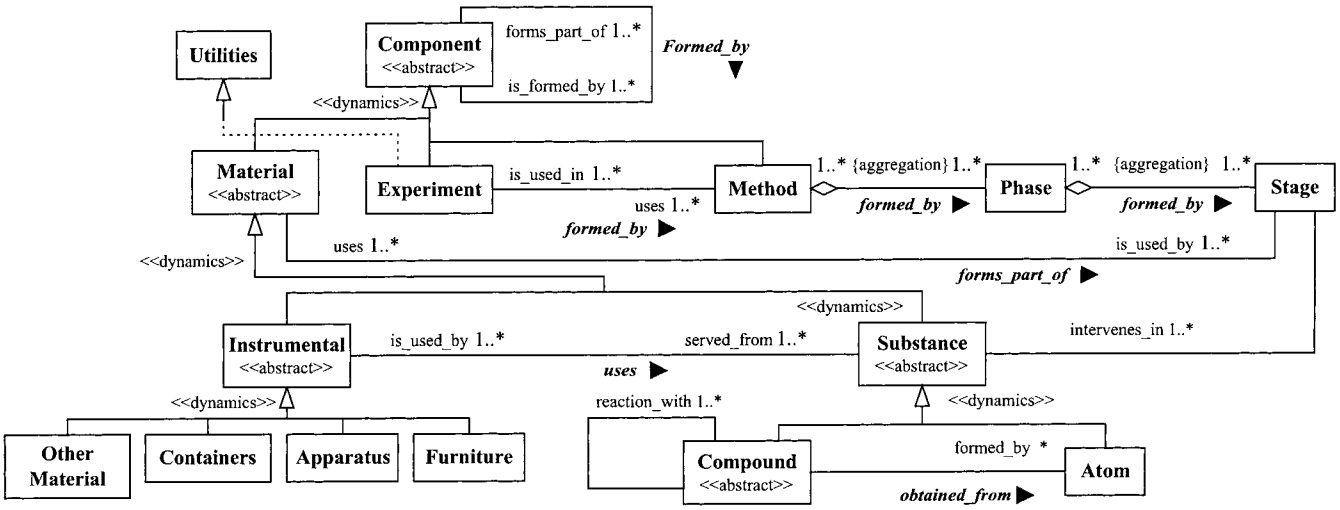

Figure 3. Class diagram of the proposed model.

relationship has been marked as nondynamic, nontransferable in ERD notation (*Entity-Relationship Diagram*), once the object has been refined its type cannot be changed.[38]

A series of components is constructed, stored in a library, and then used by a determined method, in a determined form and following a fixed timeline for the performance of an experiment, which in itself may be another component. A few examples of the objects included in this class would be the following: a decanting method, a precipitation flask, an experiment for the titration of NaOH with HCl, filter paper, a funnel, lab scales, etc.

The *Method* class is a dynamic specialization of the *Component* class and as such inherits all the attributes and methods of the *Component* generalization class. Each object contained in this class represents a procedure through which tasks are carried out in a determined order so that an experiment may be performed, thus describing a time-based view of phases, each of which comprises a series of stages in which a task is performed.

The *Phase* class is a dynamic specialization of the *Method* class. Each of the objects contained in this class represents the different parts or global steps to be completed in the course of an experiment. A phase will include one or a set of stages, its structure being determined by the level of abstraction employed in the representation.

The *Stage* class is a dynamic specialization of the *Method* class. Each object contained in this class represents the elementary functions or activities carried out in the various steps or phases employed in a given experiment following a given method. The activity represented will be as complex as the level of abstraction or detail used to represent an operation performed in the lab.

The *Experiment* class is a dynamic specialization of the *Component* class and therefore inherits all its attributes and methods. Each object from this class represents the basic lab operations which permit the discovery, checking, and demonstration of the most common chemistry phenomena, situations, or scientific principles, within a virtual environment.

The *Material* class is a dynamic specialization of the *Component* class while in turn acting as an abstract generalization class (objects belonging to the class must be refined to one of the specialization classes) of the *Instrumental* and *Substance* classes.

A Material object is characterized by its properties and functionality. Properties distinguish one material from another, in turn characterizing and identifying it. Its functionality determines how the material may be used in experiments.

The *Instrument* class is a dynamic specialization of the *Material* class and in turn acts as an abstract generalization of the *Other Material*, *Containers*, *Apparati*, and *Furniture* classes. Each object from this class represents materials commonly used in the lab.

The *Substance* class is a dynamic specialization of the *Material* class and in turn acts as an abstract generalization of the *Atom and Compound* classes. Each object contained in this class represents the basic elements of which the material is formed and which intervene in the chemical processes, mainly in reactions.

The *Utilities* class is an interface which is implemented by *Experiment* class. The *Utilities* class is in charged to represent information and methods used in the development of experiments (e.g. to obtain the molecular weight, concentration of compounds in different magnitudes, etc.).

3.2.1. Classes in the $E(V) = M + m$ Paradigm. The object classes shown in Figure 3 allow the representation of the proposed paradigm. These classes include in their definition the properties and behavior that each of the system components may exhibit through the methods.

Properties characterizing these classes may be divided into the following groups:

Identifiers: responsible for the unique identification of each of the objects from the class, both by the system at runtime and storage and by the user when accessing and loading data. For example, the identifier and name of each stage.

Descriptors: responsible for the concise description of particular class objects. The aim here is to provide the user with quick, concise, and accurate information about objects accessed. For instance, the description of a stage, danger warnings, theoretical content, and information on the material employed.

Definers: responsible for the representation of information (distinct from the foregoing kind) characterizing class objects. Their function is to store any relevant information for the

characterization of the objects employed to deliver knowledge and/or perform the virtual experiment. For example, information dealing with the obligatory nature of a step or phase, whether it is to be viewed or only used for internal operations, the dimensions and capacity of material, its location, or the composition and properties of a substance.

Executors: these represent the runtime characteristics of the objects. Designed to inform the system how each object will behave in its interaction with the user, they might include information on how to manipulate materials or valid timelines for the performance of experiments (the sequence of stages and phases, calculation of physicochemical parameters, etc.)

Visualizers: these store and show visual data for the virtual process that will accompany objects. For instance, images of the material, visual effects for the stages and phases, the appropriate virtual world, and so on.

Documentation: description of both the chemistry knowledge and the manual or automatic process accompanying the class objects. This covers the storage of aspects such as the chemistry concepts required, the experimental process, expected results, material required, or the operational process to be followed. For example: information about material, chemical substances, reactants, theoretical basis, or the extent and purpose of the experiment.

Likewise, the methods describing class behavior may be grouped as follows:

constructors: responsible for the generation of new class objects when the user defines and/or builds a new component.

destroyers: for the removal of objects that are no longer required.

modifiers: to modify object properties.

browsers: to allow access to the objects present. The aim here is to provide the user with a wide range of ways to access both the object database and memory during runtime.

executors: to perform the tasks specified in the processes defining the experiments. They should be able to run an experiment automatically as well as checking and ensuring the user carries out the tasks required of an experiment by following one of the ways defined in the experiment's task network. Executors are also responsible for object management in the virtual world using the information stored by objects in their properties along with that gained by user interaction.

Figure 4 shows, by way of example, a (partial) definition of the Stage class, indicating the various types of properties and methods described above.

4. CONSTRUCTION OF THE SYSTEM

An operational prototype of the proposed model has been built and is currently undergoing tests. It has been developed using the Java programming language, which is platform-independent and should run on any computer. The user interface has been created using Java applets that can be instantiated on most of the currently available browsers.

Information is handled by the Oracle 8i.X database manager. The object-relational features of Oracle 8i.X permits the correct implementation of the class model described in the previous section. Moreover, this DBMS allows the storage of Java and VRML procedures in the very nucleus of the database alongside SQL procedure, in addition to storing media attributes such as images and files in the

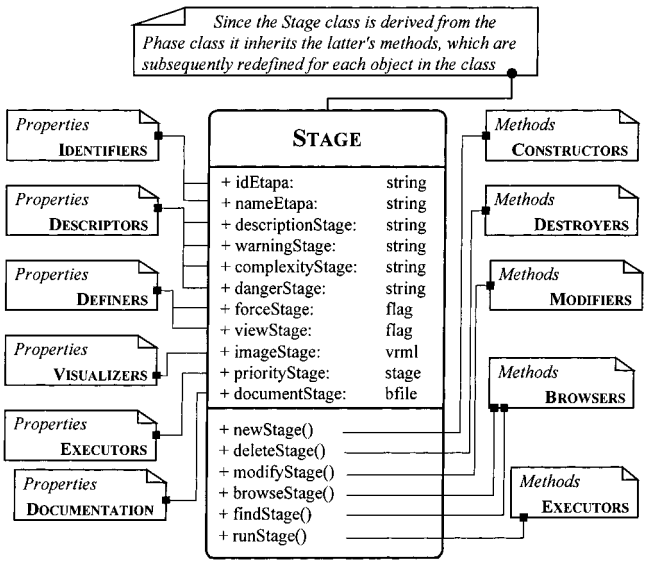

Figure 4. Property and method types in the model classes.

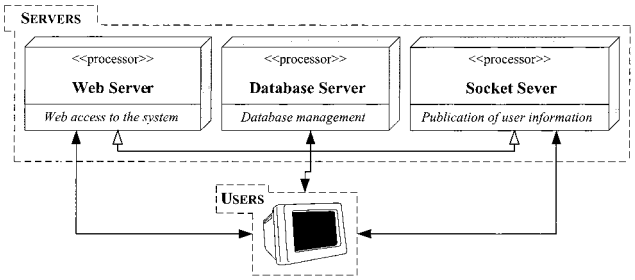

Figure 5. Deployment diagram of system statics.

definition of the table structure, which facilitates and enhances the development and performance of procedures.[39,41]

The virtual world (the labs) and its components (materials) were developed under VRML 2.0, along with utilities to permit the graphical construction of these elements and their generation in VRML format.[42,43]

4.1. System Architecture. Figure 5 shows a deployment diagram of the system statics. It may be seen that there are three differentiated layers for user interaction via any computer with a graphical interface, Java, and a VRML plug-in.[44,45]

The *Web Server* provides access to the system and its working environment; it contains the entire hierarchy of directories comprising the website.

Closely related to the *Web Server* we have the *Socket Server*, running on the same machine, and this is responsible for publishing, in the directory hierarchy of the *Web Server*, any information fed in by the user, which might be accessed by other users (e.g. VRML images of components, explanatory documentation for an experiment, etc.) The *Socket Server* is constantly listening to its clients (users) and must translate and publish on the *Web Server* relevant files that are physically resident on users' computers.

Finally, the *Database Server* is the home of all the information about lab components (containers, apparati, etc.) and of the experiments, methods, phases, and steps that have been defined. This server provides all the information required for the creation of new experiments or the running of existing ones.

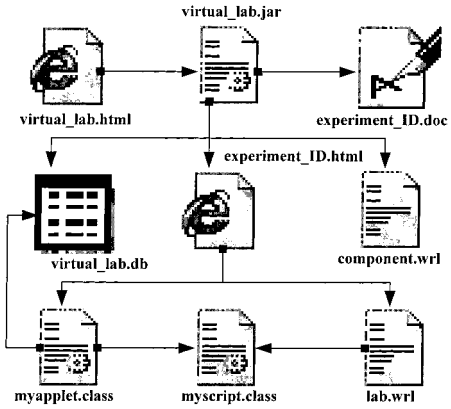

Figure 6. Deployment diagram of system dynamics.

When the user introduces a new component into the system, the corresponding information is stored on the *Database Server*, while other information connected with the component is published, via the *Socket Server*, on the *Web Server*.

A working version of the system is shown in the component diagram in Figure 6. The system starts up from a web page (*virtual_lab.html*) that contains an applet that requests user identification then runs the actual application.

All the Java classes making up the system are held in the file *virtual_lab.jar* on the *Web Server*. On managing any kind of component the user will access the system database (*virtual_lab.db*), and if the component in question has a VRML image assigned to it, this is displayed (*component.wrl*).

When a predefined experiment is run, the system launches a web page (*experiment_ID.html*), which contains an embedded VRML file containing the virtual lab associated with the experiment (*lab.wrl*). This VRML file has an associated compiled Java script listing the specific sequence of steps to be taken in order to perform the associated experiment (*myscript.class*).

Communication between the user and the virtual world is achieved through an applet (*myapplet.class*) that displays a toolbar allowing operations to be performed on the system. This applet retrieves information from the database and sends messages to the experiment's Java script detailing the actions the user wishes to carry out. The script must then control the correct execution of the experiment or show the user an error message if things go wrong.

All the documentary information connected with the experiment is linked to the *experiment_ID.html* page; links can be made automatically or manually by the user. Furthermore, this information may be accessed globally through the *experiment_ID.doc* file.

DISCUSSION

Practical chemistry teaching is in great need of further effort on the part of professionals to adapt to the present-day teaching situation, especially in undergraduate and graduate courses. Such an effort might involve the use of the new technologies and the introduction of software systems that could lead to enhanced student learning, surmounting the obstacles present in the current school environment.

The proposal detailed in this paper is based on the use of virtual laboratories to provide extra support in the teaching of chemistry laboratory experiments. The proposal is based on an $E(V) = M + m$ paradigm that permits the structuring of knowledge in chunks of information describing basic stages in laboratory usage. The grouping of these stages in phases—both stages and phases are held in a set of permitted sequences—allows definition of the method or protocol for a lab experiment to be performed.

This structure, defined using a set of transitions that are allowed on the basis of the state of the materials employed, offers the possibility of the user carrying out an experiment in any permissible order of phases and stages and is thus not restricted to just one preset sequence, as is the case with other existing systems; moreover, the system can "advise" on the most suitable sequence and give feedback on the user's errors, a feature that adds greatly to its educational value.

The model has been implemented in an operational object-oriented prototype using the latest software technology (Oracle 8i, Java, and VRML), bestowing the system with the set of properties required by authoring systems for the construction of intelligent tutorial systems; among these features are the reuse of software, easy integration of other components, and a user-machine interface suited to the needs of assisted learning.

The authors are currently working to improve and expand several of the components of the subsystems that should form part of an ITS. In particular, other systems already developed by the authors are being integrated, such as a formulator[46,47] and a balancing system for inorganic reactions.[48,49] Additionally, work is underway on the definition of a suitable ontology for the construction of a dialogue system between the user (student) and the system, such that through the use of an explanatory system running in parallel with the experiment, the student may both learn and be given explanations of conceptual and procedural aspects of the task in hand. In the present system this aspect is managed by static explanations held in.doc and.html files that are delivered to the user either on request or automatically, as might happen when some kind of exception arises.

Moreover, work is being carried out toward enhancement of the virtual world employed. Further effort is being applied to the construction of new VRML components (e.g. materials) and also to evaluating different software such as Java3D and image soundrounding for building the virtual world, hopefully reducing hardware requirements.

ACKNOWLEDGMENT

The authors would like to thank Messrs. Manuel Cachinero Olmo and Francisco Gabriel Muñoz Rodríguez for their invaluable assistance in the construction of the system, as part of their degree course at the University of Córdoba, Spain.

REFERENCES AND NOTES

(1) *Changing University Teaching: Reflections on Creating Educational Technologies*; Evans, T., Nation, D., Eds.; Kogan Page: London, 2000.
(2) Gagné, R. *The Conditions of Learning and Theory of Instruction*; New York, 1985.
(3) *Sci/Tech Librarianship: Education and Training*; Hallmark, J., Seidman, R., Eds.; Haworth Press: New York, 1998.

(4) Judson, P. N.; Fox, J.; Krause, P. J. Using New Reasoning Technology in Chemical Information Systems. *J. Chem. Inf. Comput. Sci*. **1998**, *36*(4), 621−624.
(5) Moen, E.; Boersman, K. The Significance of Concept Mapping for Education and Curriculum Development. *J. Interactive Learning Res.* **1997**, *8*(3), 487−502.
(6) Tissue, B. M. The Costs of Incorporating Information Technology in Education. http://www.chem.vy.edu/archive/chemconf97/paper04.html.
(7) Cloete, E. Electronic Education System Model. *Comput. Educ*. **2001**, *36*(2) 151−170.
(8) Lagowski, J. J. Chemical Education: Past, Present, and Future. *J. Chem. Educ*. **1998**, *75*(4), 425−436.
(9) Murray-Rust, P.; Rzepa, H. S.; Whitaker, B. J. The World Wide Web as a Chemical Information Tool. *Chem. Soc. Rev*. **1997**, *27*, 1−10.
(10) Seal, K. C.; Przasnysk, Z. H. Using The World Wide Web for Teaching Improvement. *J. Chem. Inf. Comput. Sci*. **2001**, *36*(1), 33−40.
(11) Somerville, A. N. Chemical Information Instruction in Academe: Recent and Current Trends. *J. Chem. Inf. Comput. Sci*. **1998**, *38*(6), 1024−1030.
(12) Warr, W. A. Communication and Communities of Chemistry. *J. Chem. Inf. Comput. Sci*. **1998**, *38*, 966−975.
(13) The Internet Journal of Chemistry. http://www.ijc.com.
(14) Wiggins, C. Chemistry on the Internet: The Library on your Computer. *J. Chem. Inf. Comput. Sci*. **1998**, *38*, 956−965.
(15) Borchardt, J. K. Improving the General Chemistry Laboratory. www-chenweb.com/alchem2000/news/nw-000922-edu.html.
(16) Jurs, P. C. *Computer Software Application in Chemistry*, 2nd ed.; John Wiley & Sons Inc.: New York, 1996.
(17) Rzepa, H. S. Internet-Based Computational Chemistry Tools. In *Enclyclopedia of Computational Chemistry*; Wiley: London, 1998.
(18) Tonge, A. P.; Rzepa, H. S.; Yoshida, H. Authentication of Interned-Based Distributed Resources in Chemistry. *J. Chem. Inf. Comput. Sci*. **1999**, *39*(3), 483−390.
(19) Csizmadia, F. JChem: Java Applets and Molecules Supporting Chemical Database Handling from Web Browsers. *J. Chem. Inf. Comput. Sci*. **2000**, *40*(2), 323−324.
(20) (a) marian.creighton.edu/∼ksmith/tutorials.html. (b) www.emory.edu/CHEMISTRY/pointgrp. (c) www.chem.uwimona.edu.jm: 1104/chemprob.html. (d) www.chemsoc.org.golbook.
(21) (a) chem.lapeer.org/Chem1Docs/Index.html. (b) www.geocities.com/CapeCanaveral/9687/index.html. (c) www.lanzadera.com/qgeum. (d) www.scripps.edu/∼nwhite/B/Download/. (e) jchemed.chem.wisc.edu/JCEsoft/Issues/ Series_SP/SP8/abs-sp8.html. (f) chem-www.mps.ohio-state.edu/∼lars/ moviemol.html.
(22) Barna, N.; Dori, Y. J Computerized Molecular Modeling as a Tool to Improve Chemistry Teaching. *J. Chem. Inf. Comput. Sci*. **1998**, *36*(4), 629−634.
(23) Ivancinc, O. ChemPlus for Windows. *J. Chem. Inf. Comput. Sci*. **1996**, *36*(4), 919−921.
(24) Ivanov, A. S.; Rumgantsev, A. B.; Archakov, A. I. Education Program for Macromolecules Structure Examination. *J. Chem. Inf. Comput. Sci*. **1996**, *36*(4), 660−663.
(25) Parril, A. L. Periodi 2.0 fro Macintosh. *J. Chem. Inf. Comput. Sci*. **1997**, *37*(4), 820−820.
(26) Yoshida, H.; Mausura, H. MOLDA for Windows. A Molecular Modeling and Molecular Graphics Program using a VRML Viewer on Personal Computers. *J. Chem. Software* **1997**, *3*(4), 147−156 (cssj.chem.sci.hiroshima-u-ac.jp/molda/download.html).
(27) Arruarte, A.; Fernández-Castro, I.; Ferrero, B. The IRIS Shell: How to Build ITSs from Pedagogical and Design Requisites. *Intl. J. Artificial Intelligence Educ.* **1997**, *8*, 341−381.
(28) Jonassen, D.; Reeves, T.; Hong, N.; Harvey, D.; Peters, K. Concept Mapping as Cognitive Learning and Assessment Tools. *J. Interactive Learning Res.* **1997**, *8*(3), 289−308.
(29) Murray, T. Expanding the Knowledge Acquisition Bottleneck for Intelligent Tutoring Systems. *Intl. J. Artificial Intelligence Educ.* **1997**, *8*(3), 222−234.
(30) Nussbaum, M.; Rosas, R.; Peisano, I.; Cardenas, F. Development Intelligent Tutoring Systems Using Knowledge Structures. *Comput. Educ*. **2001**, *36*(1), 15−32.
(31) Borman, S. Lab Systems Embrace the Web. *Chem. Eng. News*. January 27, **1997**, 25−27.
(32) (a) www.modelscience.com/products.html. (b) www.chemnews.com. (c) website/lineone.next/∼chemie/reviews.html. (d) www2.acdlabs.com/ilabs. (e) www.chemsw.com/10202.html. (f) www.compuchem.com/dldref/formdem.html. (g) www.ir.chem.cmu.edu/irProject/applets/virtuallab/applet_wPI.asp.
(33) Rzepa, H. S.; Tonge, A. P. VchemLab: A Virtual Chemistry Laboratory. The Storage, Retrieval and Display of Chemical Information Using Standard Internet Tools. *J. Chem. Inf. Comput. Sci*. **1998**, *36*(6), 1048−1053.
(34) Sowa, J. F. *Knowledge Representation: Logical, Philosophical and Computational Foundations*; Brooks/Cole: 1999.
(35) Luger, G. F.; Stubblefield, W. A. *Artificial Intelligence: Solutions and Strategies for Complex Problem Solving*; Addison-Wesley: Reading, MA, 1992.
(36) Jacobson, I.; Booch, G.; Rumbaugh, J. *The Unified Software Development Process*; Addison-Wesley Longman Inc.: U.S.A., 1999.
(37) Booch, G.; Rumbaugh, J.; Jacobson, I. *The Unified Modeling Language. User Manual*; Addison-Wesley Longman Inc.: U.S.A., 1999.
(38) Rumbaugh, J.; Jacobson, I.; Booch, G. *The Unified Modeling Language. Reference Manual*; Addison-Wesley Longman Inc.: U.S.A., 1999.
(39) Cary Anderson, J.; Loy Stone, B. *Manual de Oracle JDeveloper*; McGraw-Hill/Oracle Press: Madrid, 1999.
(40) Dorsey, P.; Hudicka, J. R. *Oracle8 Principios de Diseño de Base de Datos Usando UML*; McGraw-Hill: Madrid, 1999.
(41) Roehl, B.; Couch, J.; Reed-Ballreich, C. et al. *Late Night VRML 2.0 with Java*; Ziff-Davis Press: EmeryVille, CA, U.S.A., 1997.
(42) Guerrero, J. *VRML 2.0, El Lenguaje 3D de Internet*; Abeto, Madrid, 1998.
(43) Jamsa, K.; Schmauder, P.: Yee, N. *VRML, Biblioteca del Programador;* McGraw-Hill: Madrid, 1998.
(44) Lea, R.; Matsuda, K.; Miyashita, K. *Java for 3D and VRML Worlds*; New Riders: Indianapolis, U.S.A., 1996.
(45) (a) java.sun.com/products/jama-media/3D. (b) www.rtzvirtual.es. (c) www. VRML.org/Specification/VRML97. (d) www.wmaestro.com/web3d/docs/ portada.html.
(46) Luque Ruiz, I.; Cruz Soto, J. L.; Gómez-Nieto, M. A. Computer Translation of Inorganic Chemical Nomenclature to a Dynamic Abstract Data Structure. *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 4(3), 526−533.
(47) Luque Ruiz, I.; Cruz Soto, J. L.; Gómez-Nieto, M. A. Error Detection, Recovery, and Repair in the Translation of Inorganic Nomenclatures. 1. A Study of the Problem. *J. Chem. Inf. Comput. Sci*. **1996**, *36*, 6(1), 7−15. 2. A Proposed Strategy. *J. Chem. Inf. Comput. Sci*. **1996**, *36*, 6(1), 16−24. 3. An Error Handler. *J. Chem. Inf. Comput. Sci*. **1996**, 36(3), 483−490.
(48) Luque Ruiz, I.; Martínez Pedrajas, C.; Gómez-Nieto, M. A. Design and Development of Computer-Aided Chemical Systems: Representing and Balance of Inorganic Chemical Reactions. *J. Chem. Inf. Comput. Sci.* **2000**, *40*(3), 744−752.
(49) Luque Ruiz, I.; Gómez-Nieto, M. A. Solving Incomplete Inorganic Chemical Systems through a Fuzzy Knowledge Frame. *J. Chem. Inf. Comput. Sci.* **2001**, *41*(1), 83−99.

CI010015W