INFORMATION SOCIETY TECHNOLOGIES
EDUCATION AND TRAINING

SCHOOL OF TOMORROW

Project: IST-2000-25385

MODELLINGSPACE

A space for ideas' expression, modelling and collaboration for the development of imagination, reasoning and learning

User requirements

Requirements, Design Principles & Specifications document (RDPS)

Deliverable: D.01.2
Contractual Delivery Date: January 31, 2002
Actual Date of Delivery: March 31, 2002
Version: 1.0
Type: Circulation within IST Programme participants
Responsible Partner: Aegean

University of the Aegean

March 2002

ABSTRACT

The document presents the framework that allows determination of the design principles and specifications of the proposed technology-based learning environment. This document's purpose is to determine the initial requirements, the design principles as well as the initial specifications of the technology based learning environment MODELLINGSPACE. Thus, it is called **Requirements, Design Principles & Specifications** (**RDPS**) **document.** It is structured in four parts:

The first part is an introduction, situating the report in the frame of the project and explicating its purpose and status.

The second part includes the initial analysis that will inform the theoretical considerations: It analyses models and modelling in both the sciences and in education considering the modelling process as a cognitive activity. It also describes how various modelling-related concepts are incorporated in the National Curricula of a number of European countries. It presents information derived from reports on users'/ teachers' requirements of software and specially of modeling environments. Data from synchronous collaborative problem solving in class also contributes to the development of teachers' requirements. Given these requirements, the review of some significant modelling environments provides relevant research results as well as a critical analysis of the environments' features and functionalities. The subsequent review of collaborative environments provides useful information on the crucial features of these systems vis-à-vis their integration in school settings.

The third part presents the concrete theoretical considerations and requirements concerning the collaborative learning environment that will be designed and developed in the ModellingSpace project. For this purpose, some central considerations on Modelling and Collaboration are outlined and argued. These mainly concern the importance and difficulties learners experience during inquiry learning; the importance of self-regulation and support for metacognition, the importance of scaffolding; and the teachers' role in these processes. Concerning collaborative learning, the importance of using appropriate and coordinated means of actions and dialogue is highlighted. The importance of teachers' support for and supervision of students is discussed, including their analysis of students' interactions. This part takes into account the results of some informal researches that were carried out in the frame of our project in order to obtain additional requirements.

The fourth part focuses on the specifications of technology based learning environments. For this purpose it first presents typical use scenarios and learning settings. Then the design principles are outlined, expressing the requirements for the new modelling learning environment. The specifications of the environment for the main modelling and collaborative activities, as well as for the tools of analysis and support for students and teachers are listed.

Contact: **Angelique Dimitrakopoulou**
1, Democratias Avenue, 85100, Rhodes, Greece
Phone Numbers: +30 241 99127, +30 97 2880636
Fax Number: +30 241 99122
E-mail: adimitr@rhodes.aegean.gr

Main Author: **Angelique Dimitrakopoulou (University of the Aegean)**

Contributors:
Sophie Ferret (University of Angers)
George Fessakis (University of the Aegean)
Vassilis Komis (University of Patras)
Ruth Meléndez Morales (SEMA)
Argiro Petroy (University of the Aegean)
Panagiotis Politis (University of the Aegean)
Zacharoula Smyrnaiou (University of Angers)
Albert Strebelle (University of Mons-Hainaut)
Fani Stylianidou (University of the Aegean)
Vitor Duarte Teodoro (New University of Lisbon)
Annick Weil-Barais (University of Angers)

TABLE OF CONTENTS

EXECUTIVE SUMMARY

The document presents the framework that allows determination of the design principles and specifications of the proposed technology-based learning environment. This document's purpose is to determine the initial requirements, the design principles as well as the initial specifications of the technology based learning environment MODELLINGSPACE. Thus, it is called **Requirements, Design Principles & Specifications** (**RDPS**) **document**.

It is structured in **four parts**:

The **first part** is an introduction, situating the report in the frame of the project and explicating its purpose and status. The methodology applied in the identification of users' requirements and consequently of the design specifications is discussed. This is based on theoretical considerations, on research results, and involves teachers and students as users in different and multiple phases during the development lifecycle, which is thus considered an iterative process. Moreover, the design of the new open learning environment is based on 'grounded design' practice, taking into account psychological, pedagogical, technological, cultural and pragmatic considerations.

Given that the development is a grounded approach, and it constitutes an iterative process, four versions of the Requirements Design Principles and Specifications document (RDPS) document will be developed throughout the lifecycle of the project.

The present document consists the first version of the RDPS document; it is developed from theoretical analysis, previous research results and some general requirements provided by teachers, and it is considered as the *Preliminary RDPS document.*

Two software prototypes will be developed to assess the feasibility of critical factors emerging from the Preliminary RDPS.

After evaluation of the prototypes in both laboratories and real school settings, the '*Final RDPS*' will be written. The timetable for the production of the two intermediate prototypes and the Final RPDS is given.

The **second part** includes the initial analysis that will inform the theoretical considerations and will lead to the definition of the requirements. It first demonstrates how fundamental is the notion of 'model' for a number of disciplines, and how modelling itself is one of the core activities of scientific practice. It thus argues that efficient employment and management of modelling tools are key capabilities not only for the young science student, but more generally for the future citizen.

By referring to the cognitive operations which are stimulated by a modelling activity, this part then establishes that modelling, seen as a cognitive process, implies an important work of conceptualisation, both at the individual and at the social level. We then focus on computer-based modelling, examining its impact on learning and more specifically on reasoning. A special reference is made to the role of computer-based concept mapping in learning and reasoning. Overall, we claim that computer models have enormous potential for allowing pupils to explore and interact with ideas in a way that was not possible before. And this is because they can provide "both new tools for thought and new thoughts about the world, through interacting with external but artificial worlds" (Bliss, 1994a, p31).

This part of the document also describes how various modelling-related concepts are incorporated in the National Curricula of a number of European countries (mainly in the curricula of the ModellingSpace partner countries). The categories of 'models' found to be more prevalent in teaching are: 'scale models'; 'analogue models'; 'mathematical models'; and 'theoretical models'. These seem to be used in the following ways: as subject content; as visualization and didactic tools; to learn *about* science; and to learn *about* modelling.

We continue by presenting a review of the state of the art in the use and provision of computers and informatic tools in European schools, based on surveys and reports conducted and published by the European Commission. This review is seen as setting the current scene of ICT and internet usage in schools. According to it, education policies in European countries are increasingly geared to the use of ICT, which has thus become an integral part of the compulsory school curriculum in the great majority of EU countries. Modelling is not reported explicitly as one of the objectives to be pursued in the curricula; these objectives, however, cover a broad range of skills, including the use of various software packages (one of which could concern modelling) and communication via a network. The latter is still not very widespread in the EU schools, but is considered one of the high priorities of educational policies in the years to come.

Other reports on users'/teachers' requirements of software identify the need for modelling packages in a wide range of topics across the breadth of various science subjects. Primary teachers in particular make a case for modelling tools that can be usable by primary school pupils. According to all these requirements, modelling packages on the whole should: enable pupils and teachers both to explore given simulations and build their own models; be open both for demonstrations in whole-class teaching, and a context of individual/small group investigations; come with sample (pre-built) models appropriate for a given age range of pupils.

Moreover, the review of some significant computer-based modelling environments provides relevant research results as well as a critical analysis of the environments' features and functionalities, based on which a number of (general and specific) implications are drawn for the design of MODELLINGSPACE. These include the need for the new environment to allow for and facilitate:

- Modelling of different categories/classes of models.
- Concept mapping.
- Separate formalism for representation of semi-quantitative information.
- Modelling of conditional events.
- Iconic representation and animation of the model.
- Real-time interaction with the model, while it is running.
- A variety of output representations: animation, tables, graphs, numeric display of results.
- Appropriate feedback to users interacting with underlying models.
- Incorporation of scaffolding strategies.

The need for modelling environments which can be usable by younger pupils point to the use of:

- Semi-quantitative tools, since they seem to have smoother learning curves and do not require knowledge of programming or advanced mathematic skills; and
- An object-oriented framework for modelling. This seems to allow pupils to think in a kind of 'commonsense' way about the phenomena, by matching the objects and interactions that they can see in the world with what they can model in the modelling environment, rather than have to translate them into abstract representations or variables.

This second part of the document concludes with a critical review of technology-based collaborative learning environments. This provides useful information on the crucial features of these systems vis-à-vis their integration in school settings. Some of their implications for the design and development of MODELLINGSPACE's collaborative environment are:

- Control of collaborative interactions: The system must not act simply as a vehicle for collaboration. It must be involved in helping co-ordinate the collaboration, facilitating the negotiation and the whole process of problem solving by: structuring the dialogue among pupils; structuring public workspaces; providing guidance for collaborative learning interactions.

- Awareness of collaboration: If we want pupils to be able to work collaboratively, they must be aware about other participants' actions. In order to enable this awareness, the system must provide relevant information such as other learners' locations in the shared workspace, their actions, the interaction history.

Support for teachers: In order to allow teacher to detect when particular children are having difficulties, to identify skills which seem to be generally weak across a group of children and also to be able to suggest when a group of children are not working well together, the system must provide information about the task each learner works on, attendance of every learner's private workspace, attendance of the public workspace, elaborated comparative information concerning various workspaces.

Finally, this part presents the main elements of a previous prototype 'ModelsCreator' that was implemented in Greek schools, as well as some main users' requirements for the environment's improvement and extension.

The third part presents and argues ModellingSpace's central requirements and considerations related to modelling activities and collaboration in real school settings. For this purpose it takes into account the analysis of the previous part and particularly teachers' general software requirements, national educational policies and curricula, the theoretical framework which arises from research results and the review of existing systems. Additionally, it considers and makes references to the results of informal research that was carried out in the frame of ModellingSpace project, in order to explore specific points (related documents are presented in the Annexes (A'-E')).

In its first sub-part, it explicates the main reasons for conceiving and designing a new modelling environment. It specifies requirements related to inquiry learning and modelling support. It points out the importance of self-regulation, scaffolding and metacognitive support. Finally it argues about the teachers' roles and needs during modelling activities in school settings.

The second sub-part is devoted to the ModellingSpace's envisaged support to collaborative learning and to the learning community. It firstly presents explicitly the reasons why ModellingSpace project focuses on collaborative learning and the creation of a learning community. Then, it highlights the importance of appropriate means of action, coordination and awareness, as well as the importance of appropriate means of dialogue during synchronous and/or asynchronous collaborative learning activities. It focuses on the role and needs of teachers in order to manage these activities as well as on the necessity to provide specific analysis tools for teachers. Finally, it presents our vision of the emerging technology-based learning community and of how to support it.

The fourth part focuses on the specifications of technology based learning environments. For this purpose it first presents typical use scenarios and learning settings. Then the design principles are outlined, expressing the requirements for the new modelling learning environment. The specifications of the environment for the main modelling and collaborative activities, as well as for the tools of analysis and support for students and teachers are listed.

PART I

INTRODUCTION

The Introductory Chapter of DO1 describes the general purpose and objectives of the ModellingSpace project and gives the context and status of this document in the framework of the project as a whole.

1. The Document Context: ModellingSpace Project and its main objectives

ModellingSpace IST-2000-25385 is an IST project, part of the Education and Training program's Action Line: School of Tomorrow. The ultimate goal of ModellingSpace is to develop an open learning environment, which permits and support modelling activities by young students, and to implement it with alternative pedagogical strategies in real school settings.

In more detail, ModellingSpace project aims to:

⇨ Develop an open modeling environment that support collaborative problem solving in real school settings:

MODELLINGSPACE is an open learning environment, which enables students to create and work on different categories of models (semantic models as concept's maps and logic formalisms, semi-quantitative and quantitative models). In order to support them, it offers rich visualisation, real worlds simulations, alternative and multiple forms of representations and meta-cognitive tools.

It will contain several main components permitting multimedia presentation of situations under study, and the design, testing and validation of models. These components will include tools for model building (entities and relations), representations, meta-analysis, etc..

Its purpose is to support and teachers during learning/teaching activities. Tools will be developed in order to support collaboration between students as well as to provide assistance to teachers. Examples of the latter will include awareness of others activity tools, collaboration dialogue based tools, and automated comparison of on-line students work.

⇨ Conceive appropriate learning activities for:

- a wide range of students (11-17 years old), following different curriculum subjects in a variety of forms of class organisation in a number of European countries;
- modeling situations studied in mathematics, physics, chemistry, biology and environmental education, related to the existing national curricula and other interdisciplinary situations.

⇨ Determine appropriate pedagogical approaches, teaching strategies and tactics for both inquiry learning through modelling activities as well as for learning through fruitful collaborative problem solving, based on research in real school settings.

⇨ Provide appropriate accompanying material for both teachers and students in 4 European languages (English, French, Portuguese, Greek).

⇨ Educate and support teachers in innovative teaching and learning strategies and, through the creation of a community of learners, contribute to the promotion of a community of practices.

⇨ Contribute to research on collaborative learning and teaching using ICT in schools and at a distance. The research during system development will be carried out in primary and high schools in national levels and will be focused on multinational application of collaborative distance learning and teaching in high schools. The methodological paradigm is based on ethnographic (systemic) and discourse- analytic methods.

The project is realized by a consortium of Learning Technology and Educational Engineering Laboratory of the Department of Education, Aegean University (GR), the HCI group of University of Patras, the New University of Lisbon (Pt), the University of Mons-Hainaut (Be), the Laboratory of Psychology, University of Angers (Fr), and Schlumberger SEMA (S).

2. Document Purpose

The purpose of a Users' Requirements document is to determine the list of characteristics of the system that developers will build within the project.

"Users' requirements'' is usually the document that lists the requirements of typical users according to specific engineering methodologies. This is the typical case when the objective is to develop a technology-based environment that is addressed to adults, for professional or educational reasons.

This document (called Requirements Design Principles and Specifications (RDPS)) develops the common practice of design of innovative learning environments to provide a basis tailored to the specific objectives of ModellingSpace. The document aims to:

- determine the requirements on the basis of theoretical considerations, analysis of educational policies, review of existing environments and requirements provided by users;
- define the main design principles and the initial specifications of the environment;
- verify and validate some main assumptions that were taken in the project on which ModellingSpace is built – ModelsCreator.

Thus, the purpose of this document is to present the various analyses that will result to determine not only the Requirements, but also the Design Principles as well as the initial specifications, and so from now on is called "**Requirements, Design Principles & Specifications**" (**RDPS**) **document.**

The RDPS does not contain requirements concerning learning activities or pedagogical approaches. Learning activities will be presented in the deliverable D05, while pedagogical approach will be a part of the deliverable D02.

3. Document Status

The RDPS document shall discuss and define *what* is to be developed in the project, but not *how* this will be done. In particular it does not include any detailed design, verification or project management details.

Methods and procedures to check or validate the final application of the project against pedagogy or usability will be defined within the research report DO2 (version 1, internally circulated on 30 September 2002). The final version will be determined and delivered as part of the official DO2 document at the end of December 2002.

Concerning the intended audience of this document, it has to be noted that the document is intended to participants of the ModellingSpace project as well as of the IST programme.

4. Users' Requirements and Methodology of Learning Environment Design Principles and Specifications

When we have to develop software for adults to use in a professional or everyday setting we apply the typical approaches and methodologies of users' requirements engineering. In general, and most of the times, these approaches are based on the following basic considerations: the users themselves will provide directly the requirements which will be negotiated and finally defined jointly by the developers taking into account project objectives and project available costs and time.

In the case of learning environments addressed to primary and secondary education the 'users' are typically the 'students', 'teachers' and 'curriculum developers, and consequently:

- It is not possible to extract initial requirements in order to design a new learning environment by direct questions to children.
- It is not possible to extract initial requirements directly from teachers when an innovative learning and teaching environment is to be designed. Such approaches applied in the early stages of educational technology development were being disappointing (Zucchermaglio 1992, Dimitracopoulou 1998), for multiple reasons, such as: they lead to the development of systems that reproduce current practices (e.g. drill and practice); they do not take into account recent issues from science education or cognitive psychology on the way that children learn and the difficulties that they have. These systems often induce misconceptions in students via non-studied representations systems, etc.

Consequently, in the case of technology-based learning environments addressed to young students:

- The methodology that leads to the specification of design principles and learning environment specifications, is more complex, based on theoretical considerations, on research results, and involves teachers and students as users in different multiple phases during the development lifecycle (not in the initial phase).
- The initial concept and the design principles are not defined by users, or by developers, not even by both of them, but by a whole interdisciplinary group that is composed of teachers involved in the development project together with specialists in the learning of the appropriate subject matter (science education, mathematics education, etc); cognitive psychology; the design of technology-based learning environments of the appropriate kinds (the design of a modeling environment is different from the design of a hypermedia system or this of a robotics), human computer interaction for educational software, developers.

Open-ended innovative learning environments are based on "Grounded Design". Grounded Design is "the systematic implementation of processes and procedures that are rooted in established theory and research in human learning" (Hannafin, Land and Oliver 1999, p.102, Land and Hannafin 2000).

We discuss four conditions as basic to grounded design practice:

First, the design must be rooted in a defensible and publicly acknowledged theoretical framework.

Next, methods must be consistent with the output of research conducted to test, validate, or extend the theories upon which they are based.

In addition, grounded designs are generalisable, that is they transcend the individual instances in which isolated success may be evident and can be adapted or adopted by other designers. This does not suggest a literal algorithmic map according to strictly defined conditions but rather the heuristics-based application of design process appropriate in comparable circumstances.

Finally, methods are proven effective in ways that support the theoretical framework itself as successive implementations clarify the approach. The design processes and methods continuously inform, test, validate, or contradict the theoretical framework and assumptions upon which they are based, and vice-versa.

Learning environments are rooted in five core foundations: psychological, pedagogical, technological, cultural, and pragmatic. (Hannafin and Land,1997).

- Psychological foundations emphasize theory and research related to how individuals think and learn.
- Pedagogical foundations form the affordances and activities of the environment and should be inextricably linked to corresponding psychological and epistemological foundations.
- Cultural foundations reflect the prevailing values of a learning community. For instance, cultural foundations may reflect particular values such as 'interdisciplinary learning', or 'global society';
- Technological foundations influence how media can support, constrain or enhance the learning environment. A variety of media can be exploited to support learning in a variety of ways, but grounded deployment of technology is linked to the particular epistemological frame in particular ways. Technology foundations determine that which is technologically possible, but grounded practice requires determination of how these capabilities should be exploited.
- Finally, pragmatic foundations emphasize the reconciling of available resources and constraints with the actual design of any given learning environment. Many schools, for instance, perceive the benefits of connecting teachers, students, administrators to others across the world. Limitations in connectivity, bandwidth and hardware, however, often limit what can be accomplished pedagogically and technologically. Pragmatic foundations represent the reality check of learning environment design and implementation, frequently causing a reassessment of alignment among one or more foundations.

Grounded design, therefore, involves the simultaneous alignment of each foundation in order to optimise coincidence across all foundations.

The grounded design aligns the foundations, assumptions and methods and identifies main principles underlying design that will guide the determination of the software specifications. Additionally, in the RDPS we point out to research results of other projects on modelling and learning in order to assure requirements and specifications (in parallel with DO2 report which is partially written).

The analysis presented in this RDPS document is based on dimensions/factors that we consider necessary in order to determine the design principles and characteristics of a computerized learning environment. These include:

- the technological environment: the methods, approaches and techniques of other, previously constructed environments, as well as the evaluation output from the utilization of these systems by the students;
- the learner him/herself: to analyze their general characteristics, difficulties, the intuitive ideas of children of the target age group concerning the specific subject area and learning activities;
- the subject matter and the specific learning activities: to proceed to an epistemological analysis of content as well as a cognitive analysis of specific tasks and learning activities (e.g. which cognitive operations are involved in each task);

- the social context: here we distinguish between a closer and a broader social environment. In the closer social context we include the teacher, the learner, the other students of the class, and the conditions of the class work and interactions. These lead us to make explicit assumptions about the learning process; we derive from them a number of pedagogical principles. In the wider social environment we include the whole school (status of the school, such as an 'innovative' or 'traditional one'), the national educational policy on ICTs, the recent and previous curricula, etc..

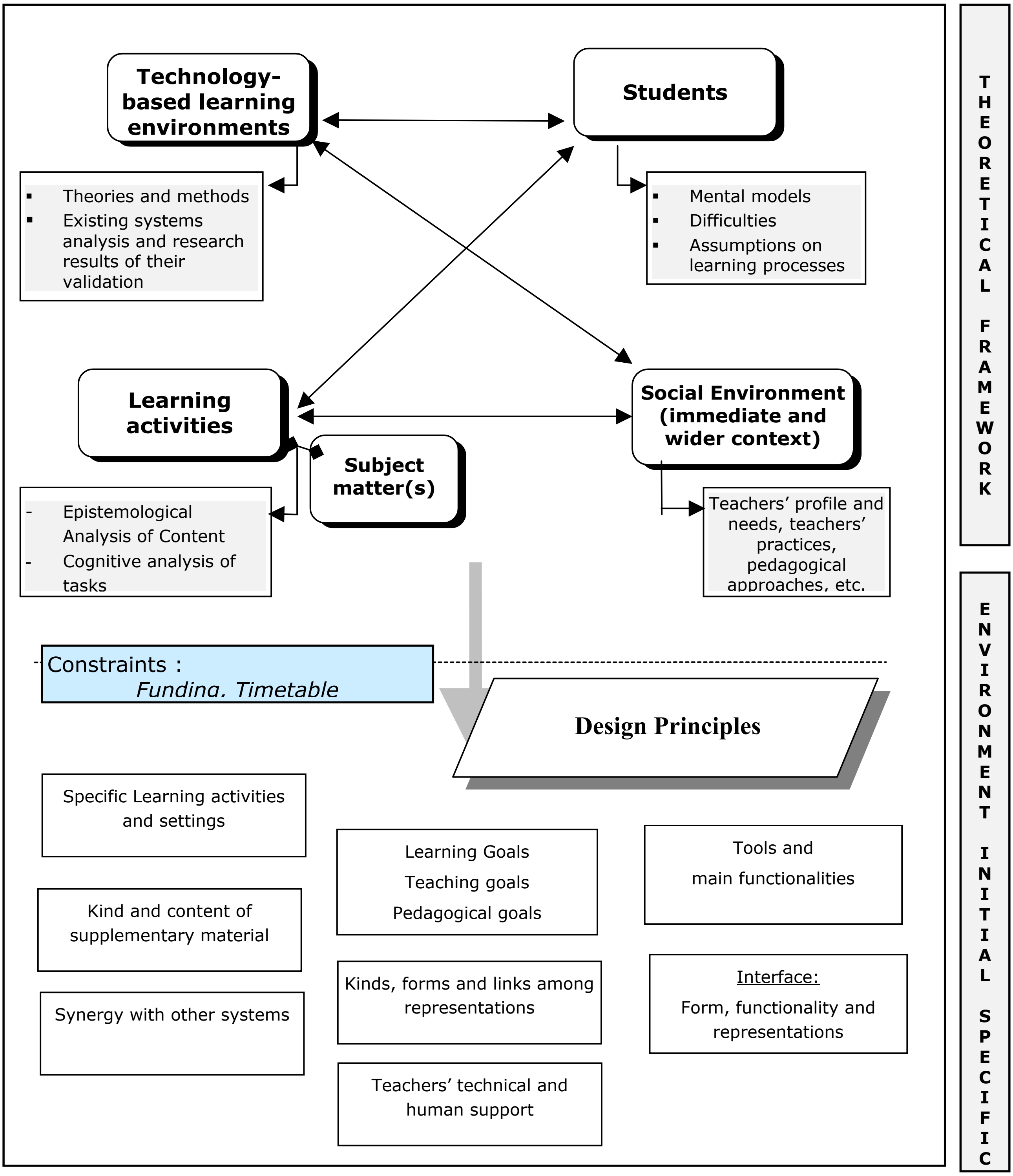


Figure 1.1: Relationship between the theoretical framework and the learning environment's initial specifications

The analysis of each dimension, as well as the interrelations between them, provide a basis for us to make decisions specifying the characteristics of the technological environment; the tasks and learning activities; and the pedagogical approach adopted (See Figure 1.1).

5. Timetable for release of Design Principles and Learning Environment Specifications

Given that the development is an iterative grounded design, four versions of the RDPS document will be developed throughout the lifecycle of the project (see Figure 1.2.)

The first version of the RDPS document is developed from theoretical analysis, previous research results and some general requirements provided by teachers, as well as some requirements that result from informal experimentations organized in the frame of ModellingSpace project, in order to explore specific aspects, not well documented until now. This document is called *Preliminary RDPS* document. (This consists the present D01 deliverable).

Two prototypes will be developed to assess the feasibility of critical factors emerging from the Preliminary RDPS.

After evaluation of the prototypes in both laboratories and real school settings, the '*Final RDPS*' will be written. The timetable for the production of the two intermediate prototypes and the Final RPDS is given below.

Timetable of Requirements, Design Principles and Specifications Document (RDPS)[1]:

DO1.v2. Intermediary version: First release of modelling tools, taking into account requirements provided by teachers and students

DO1.v3. Intermediary version: Second release of modelling tools, and first release of collaborative tools and functionalities as defined by requirements provided by teachers and students

DO1.v4. Final RDPS: final version of the whole MS environment Design Principles and Specifications, developed from students requirements, teachers requirements and research results

Consequently, each of the intermediary versions (DO1.v2, and DO1.v3) will embody changes to overcome perceived shortcomings or inaccuracies in the previous version of the RDPS. In addition, it is expected that the development of the learning environment will lead to the emergence of additional needs – especially from teachers and students – and these will also be incorporated into appropriate versions of the RDPS. Each and every new release will be submitted to the project committee for acceptance. The environment will be stabilized through the learning scenarios, the students and teachers' most appropriate settings and roles.

We need to note here that the intermediary RDPS2, and RDPS3 will contain only the parts that are susceptible to changes, thus Part IV (Design principles and specifications) of the current document.

[1] The above RDPS releases were defined as internal documents to the project partners, and they are not explicitly indicated to the Technical Annexe. However, it is possible for them to be made available to the project officer and project reviewers, if required.

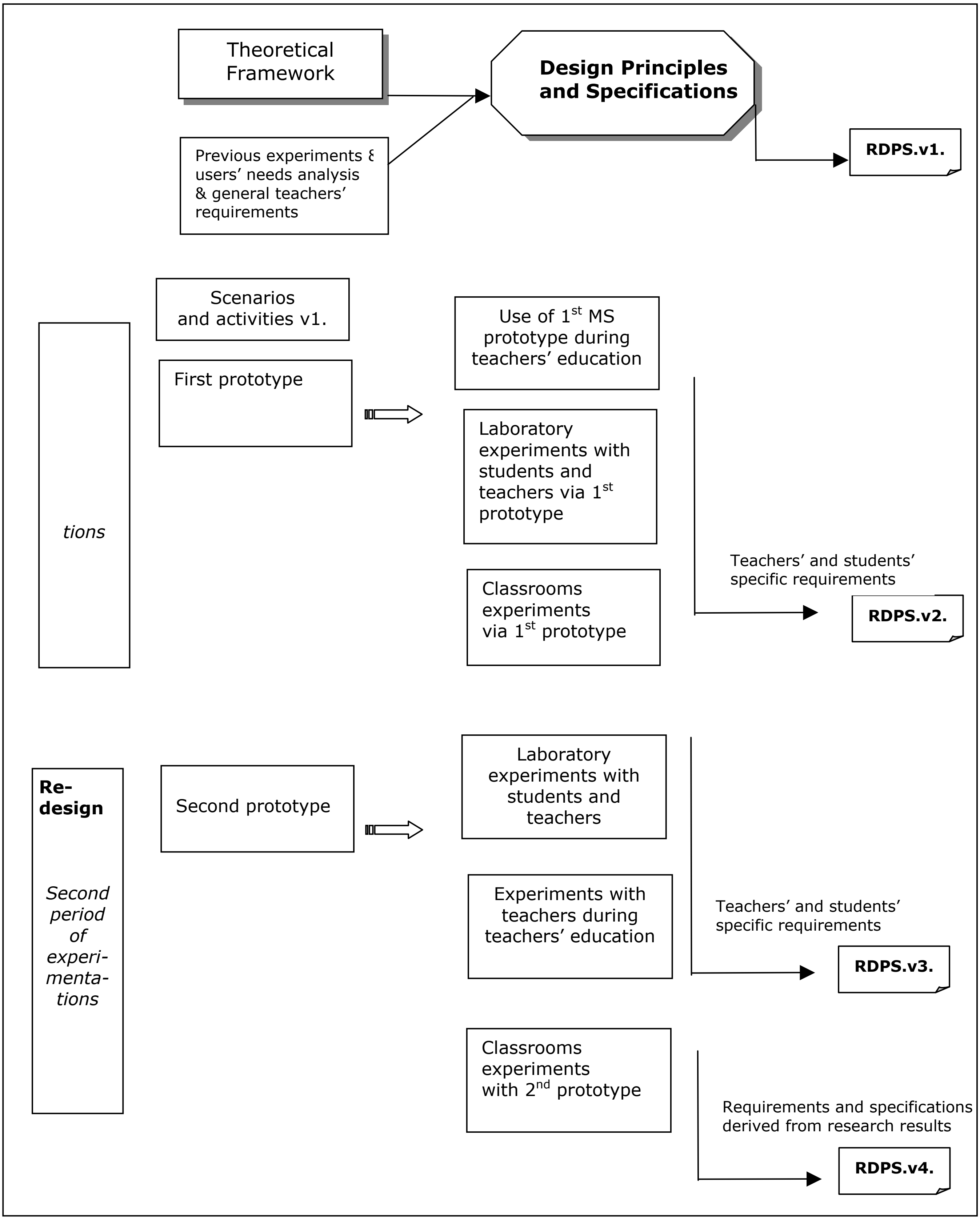


Figure 1.2: RDPS versions and experimentation process

6. RDPS Document structure

The rest of the document is structured in the following way:

The second part presents the analysis that will define the theoretical considerations, necessary for the design principles and specifications of the initial technology-based learning environment:

- This part analyses models and modelling in both the sciences and in education considering the modelling process as a cognitive activity.
- It also describes how various modelling-related concepts are incorporated in the National Curricula of a number of European countries.
- It presents information derived from reports on users'/ teachers' requirements of software and specially of modeling environments. Data from synchronous collaborative problem solving in class also contributes to the development of teachers' requirements.
- Given these requirements, the review of some significant modelling environments provides relevant research results as well as a critical analysis of the environments' features and functionalities. The subsequent review of collaborative environments provides useful information on the crucial features of these systems vis-à-vis their integration in school settings.
- Finally, it presents briefly the previous ModelsCreator prototype as well as related requirements for improvement.

The third part presents concrete considerations on the collaborative learning environment that will be designed and developed in the ModellingSpace project.

- For this purpose, some central considerations on Modelling and Collaboration are outlined and argued. These mainly concern the importance and difficulties learners experience during inquiry learning; the importance of self-regulation and support for metacognition, the importance of scaffolding; and the teachers role in these processes.
- Concerning collaborative learning, the importance of using appropriate and coordinated means of actions and dialogue is highlighted. The importance of teachers' support for and supervision of students is discussed, including their analysis of students' interactions.

The fourth part focuses on the specifications of technology based learning environments.

- It presents typical use scenarios and learning settings.
- The design principles are outlined, expressing the requirements for the new modelling learning environment.
- The specifications of the environment for the main modelling and collaborative activities, as well as for the tools of analysis and support for students and teachers are listed.

Finally, the annexes present documents related to some investigations carried out in the frame of ModellingSpace project in order to determine appropriate requirements and specifications related to aspects that they are not actually well explored by the international literature (concerning mainly aspects of collaboration, and needs and requirements related to teacher support).

7. References

Dimitracopoulou, A. (1998) Designing Educational Software: From the empirical approaches in an interdisciplinary approach. *Journal of Contemporary Education,* Part A'., No 100, pp. 114-123 & Part B': No 101, pp. 95-103.

Hannafin, M.J. and Land, S. (1997) The foundations and assumptions of technology-enhanced, student-centered learning environments. *Instructional Science,* 25, pp 167-202.

Hannafin, M.J., Land, S. and Oliver, K. (1999) Open Learning environments: Foundations, methods, and Models. In C. Reigeluth (Ed). *Instructional design theories and Models* (vol.2.) (Mahwah, NJ.: LEA).

Land, S. and Hannafin, M.J. (1997) Student centered learning environments. In D. Jonassen and S. Land (Eds). *Theoretical foundations of Learning Environments*, LEA.

Zucchermaglio, C. (1992). A methodological approach for a cognitive design of technological learning environment. Paper presented at the NATO, Advanced Study Institute on '*Psychological and Educational Foundations of Technology-Based Learning Environments',* July 26-August 3, 1992, Crete, Greece.

PART II

THEORETICAL CONSIDERATIONS AND REVIEW OF EXISTING TECHNOLOGY-BASED LEARNING ENVIRONMENTS

1. Models and Modelling in Sciences

1.1 Introduction

> *The sciences do not try to explain, they hardly even try to interpret; they merely make models. By a model is meant a mathematical construct which, with the addition of certain verbal interpretations, describes observed phenomena. The justification of such a mathematical construct is solely and precisely that it is expected to work.*
>
> *- John von Neumann [Neumann, 1947, pp:180-196]*

The concept of 'scientific model' is a key one, common among many (if not all) disciplines. Consciousness of this concept enables science exploration from an advantageous point of view. General models store abstract scientific knowledge. Detecting common, similar or comparable models among different disciplines accelerate knowledge transfer and reusability [Bertalanffy, 1969, p80].

Traditionally scientific disciplines develop and employ different representation systems for their models. These representations could be seen as inscription systems that are optimised for the corresponding rhetoric and thinking [DiSessa, 2000, pp: 7, 18, 27, 32, 35]. Currently, information technology supports old representation systems (e.g. algebra, differential calculus) extending their applicability and simplifying their exploitation and management. Furthermore, information technology provides new representation systems common among various disciplines. Some of these new representation systems (e.g. spreadsheets, databases etc) are used even in everyday problem solving activities. The impact of these new representation systems is very important but not completely assimilated by scientists and/or education systems.

Efficient employment and management of modelling tools are key capabilities for the young science student and the future citizen.

1.2 Use of the concept of 'model' in scientific practices

The notion of 'model' appears in mathematics logic as a well-defined concept. It is also used in other sciences without formal definition. In the followings we will present the general scientific use of 'model' and then its mathematical definition. Finally we will try to locate relations between the two uses. It will be argued that the general scientific meaning of the 'model' concept corresponds to the concept 'theory' of mathematics logic.

1.2.1 The general scientific use of the 'model' concept

We examined the notion of 'model' as this is portrayed in science textbooks and papers of different. Textbooks usually contain stable scientific views and young students contact science manly by using them. Scientific textbooks are a quite reliable information source for the current use of the notion. A characteristic scripture follows:

Introduction

Inferring models from observations and studying their properties is really what science is about. The models ("hypotheses", "laws of nature", "paradigms", etc.) may be of more or less formal character, but they have the basic feature that they attempt to link observations together into some pattern. System identification deals with the problem of building mathematical models of dynamical systems based on observed data from the systems. The subject is thus part of basic scientific methodology, and since dynamical systems are abundant in our environment, the techniques of system identification have a wide application area.

…

1.2 Models

Model types and their use

When we interact with a system, we need some concept of how its variables relate to each other. With a broad definition, we shall call an assumed relationship among observed signals a model of the system. Clearly, models may come in various shapes and be phrased with varying degrees of mathematical formalism. The intended use will determine the degree of sophistication that is required to make model purposeful.

No doubt, in daily life many systems are dealt with mental models, which do not involve any mathematical formalism at all. To drive a car, for example, requires the knowledge that turning the steering wheel to the left induces a left turn, together with subtle information built up in the muscle memory. The importance and degree of sophistication on the latter should of course not be underestimated.

For certain systems it is appropriate to describe its properties using numerical tables and/or plots. We shall call such descriptions graphical models. Linear systems, for example, can uniquely described by the impulse or step responses or their frequency functions. Graphical representations of these are widely used for various design purposes. The nonlinear characteristics of, say, a valve are also well suited to be described by a graphical model.

For more advanced applications, it may be necessary to use models that describe the relationships among the system variables in terms of mathematical expressions like difference or differential equations. We shall call such models mathematical (or analytical) models. Mathematical models may be further characterized by a number of adjectives (time continuous or time discrete, lumped or distributed, deterministic or stochastic, linear or nonlinear. Etc) signifying the type of difference or differential equation used. The use of mathematical models is inherent in all fields of engineering and physics. In fact, a major part of the engineering field deals with how to make good designs based on mathematical models. They are also instrumental for simulation and forecasting (prediction), which is extensively used in all fields, including no technical areas like economy, ecology, and biology.

The model used in a computer simulation of a system is a program. For complex systems, this program may be built up by any interconnected subroutines and lookup tables, and it may not be feasible to summarize it analytically as a mathematical model. We use the term software model for such computerized descriptions. They have come to play an increasingly important role in decision making for complicated systems.

This scripture is from the introductory chapter of a textbook titled *«System Identification: Theory for the user»* [Ljung, 1987]. This subject concerns mainly computer scientists, physicists, mathematicians, and electrical and electronics engineers.

The scripture considers modelling a central scientific activity and the model construction the main goal of science: *"inferring models is really what is science about".* A model describes the relationships among the variables of systems. Scientific models are expressed as numerical tables and/or plots (graphical models), sets of algebraic, differences or differential equation systems (mathematical models), and complex control and data structures (software models). Models are developed for designing purposes in engineering; forecasting is another use of models. They appear even in non-technical disciplines like economy, ecology and biology. The author uses the notions 'model', 'hypothesis', 'law', and 'paradigm' interchangeably.

1.2.2 The concept of 'model' in mathematics logic

The term 'model' is used in the framework of languages and theories of mathematics logic [Anapolitanos, 1985, pp:163-183]. 'Model' is a term technically defined in mathematics logic. It relates naturally with the concept of 'theory'. A first order theory is a first order language with the addition of axioms. These axioms could be arbitrary defined but often are defined according to a real world system, the behaviour of which is used to validate the theory.

Mathematics logic is related to scientific modelling and this relationship can be described more easily through the use of Artificial Intelligence (AI) as knowledge representation system [Rich and Knight, 1991; Ginsberg, 1993].

Prolog (Programming in Logic) is a well-known programming language [Bratko, 1990, pp:63-64] that is used for the construction of knowledge models using the mathematics logic as a representation system. Using Prolog the programmer or Knowledge engineer makes use of formal logic in order to construct a knowledge base for the description of a domain (system, or world). This process is basically a process of selective abstraction, similar to the modelling process. Furthermore the knowledge base that is produced is used for similar purposes to usual models in the context of problem solving.

1.2.3 'Theories' vs. models

Constructing a 'theory' in Logic is basically the same process as modelling in any other representation system. The modeller uses abstraction in order to describe a real world subsystem. The modeller has to choose what is going to be a fact, what is going to be a relationship, and which axioms (rules) are going to be defined. The 'theory' is then evaluated by comparison of its use products with the real world system.

Theories in Mathematics Logic are often abstract and their axioms are products of imagination, in other worlds there is not a real system corresponding to them. Pure mathematical 'theories' are evaluated only for their logical consistency and completeness and not for their utilization according to a real world system.

With procedural representations scientists have to describe precisely the solution processes and there is not a standard universal inference machine. Instead of a procedure specification for a precise solution, in Logic models there are theorem provers that reduce problem solving to searching in problem's state space.

1.3 Towards a definition of 'scientific model'

Minsky defined the 'model' concept and examined its relationship to knowledge in the work titled "Mater, Mind, and Models" [Minsky, 1968]. The following scripture contains the definition he gave and is representative of Minsky's point of view:

If a creature can answer a question about a hypothetical experiment without actually performing it, then it has demonstrated some knowledge about the world. For, his answer to the question must be an encoded description of the behaviour (inside the creature) of some sub-machine or "model" responding to an encoded description of the world situation described by the question

We use the term "model" in the following sense: To an observer B, an object A* is a model of an object A to the extent that B can use A* to answer questions that interest him about A.

The model relation is inherently ternary. Any attempt to suppress the role of the intentions of the investigator B leads to circular definitions or to ambiguities about "essential features" and the like. It is understood that B's use of a model entails the use of encodings for input and output, both for A and for A. If A is the world, questions for A are experiments. A* is a good model of A, in B's view, to the extent that A*'s answers agree with those of A's, on the whole, with respect to the questions important to B. When a man M answers questions about the world, then (taking on ourselves the role of B) we attribute this ability to some internal mechanism W* inside M. It would* *be most convenient if we could discern physically within M two separate regions, W* and M-W*, such that W* "really contains the knowledge" and M-W* contains only general-purpose machinery for coding questions, decoding answers, or administering the thinking process. However, one cannot ready expect to find, in an intelligent machine, a clear separation between coding and knowledge structures, either anatomically or functionally, because (for example) some "knowledge" is likely to be used in the encoding and interpreting processes. What is important for our purposes is the intuitive notion of a model, not the technical ability to delineate a model's boundaries. Indeed, part of our argument hinges on the inherent difficulty of discerning such boundaries.*

Based on the above scripture, we conclude that model M is an idealized representation (or expression, or description) of a physical system S produced by abstraction, which potentially produces behaviour proportional to S.

1.4 Uses of models

Models are used for:

- Expression

Different modellers often produce different models for the same physical system. A model thus expresses the modeller’s ideas, knowledge, and even beliefs (like “perfect shapes” and similar ideas held by Kepler and others) about a physical system.

- Communication

Models are materials that can be used for communication purposes. A model can be used to store knowledge for future use (information transfer in time) by other people and can also be transferred in space carrying messages to people in other places. A model could be used for the externalisation of a modeller’s thoughts in order that they be visible and explicit for colleagues, teachers, students, friends etc. in the context of a dialogue.

- Forecasting – prognosis

Forecasting is one of the most known uses of models. The simulation (often mentioned as ‘running’) of a model for certain selection of model parameters produces a series of results that can be used for the prediction of the behaviour of physical systems under analogous conditions. As an example, a motion model can be used for the estimation of the time needed for an object to fall from a known height building without actually throwing the object.

- Explaining

Explanation is the identification of the conditions that could be responsible for the production of a given behaviour for a physical system. The explanation production seems reverse in comparison to forecasting. For example, it would be interesting to find out the mechanism of a building collapsing after an earthquake or the conditions under which a bullet could reach a certain target.

- Understanding

The expression of ideas, hypotheses etc. in the form of a dynamic model gives the opportunity to compare the consequences of the model with the physical system’s behaviour. Every mismatch between the model and the physical system can result in resolving a misconception. Consciousness of misconceptions is the first step for understanding development; debugging or modification of the model can be the indication of a better model adoption by the modeller.

Scientists and engineers often build models in order to understand complex systems [Conrad and Mills, 2000].

- Optimisation

Optimisation is a problem domain with increasing interest that refers to problems with intractable complexity (NP-Complete and NP-Hard problems) [Cormen, Leiserson and Rivest, 1991]. There are interesting optimisation problems for which analytical end even algorithmic solutions are just not applicable or efficient. These problems are dealt with often using heuristic solutions. In this manner a model can be used in order to optimise a system or estimate an optimal decision using simulation to compare alternatives.

1.5 The relationship to physical system and real world

For the examination of the relationship of models to real world the notion of 'physical system' is used. A 'physical system' is an internal conceptual construct that an observer builds by perception of and interaction with the external world. Models are external descriptions of physical systems. The approach that is presented here shares mainly the view of Kirbach and Schmidt as it was described in the paper titled *"On the use of models in education"* [Kirbach and Schmidt, 1976].

Analysing the physical system and model construction processes we can describe three levels of correspondence between models and physical systems (Figure 1.1):

The semantic correspondence level: This level concerns relationships between the physical system's conceptual framework and the model's representation system. (E.g. In a model of motion the concept 'speed' in the physical system's conceptual framework would correspond to the concept 'variable' in the algebra's representation system or to the concept 'vector' in analytic geometry).

The syntactic (or structural) correspondence level: This level concerns relationships (in structural terms) between the physical system and the model's expression according to the syntactic rules of the representation system. (E.g. A specific lever of a machine corresponds to a specific line on a technical drawing, or a machine motor to a repetition construct of an algorithm.)

The pragmatistic correspondence level: This level concerns relationships among operations on the physical system and the model. The term operation includes real and cognitive actions. By interacting with the model and the physical system the user develops pragmatistic relations among them.

In the modelling side of Figure 1.1 there is the conceptual framework of the representation system. Conceptual systems form the abstractions within which observed information becomes meaningful. Semantic correspondence is established between the physical system's conceptual framework and the model's representation conceptual framework. A modeller implements models using the representations' conceptual framework. The more easily a semantic correspondence can be established the easier a model is produced for a physical system with the specific representation system.

The syntactic correspondence is established between the realization of the model and the physical system.

When the user interacts with a model and relates the outcomes with the real world s/he is constructing the pragmatic correspondence between model and real world.

The sharing by different observers of the correspondences between real world and model is very important in order for the observers to agree upon what constitutes the real while using a model. This means that a learner may create misconceptions when s/he uses a model to try to understand a real system. It is important that learner has access to the real system or a good construct of expertise before the use of its model.

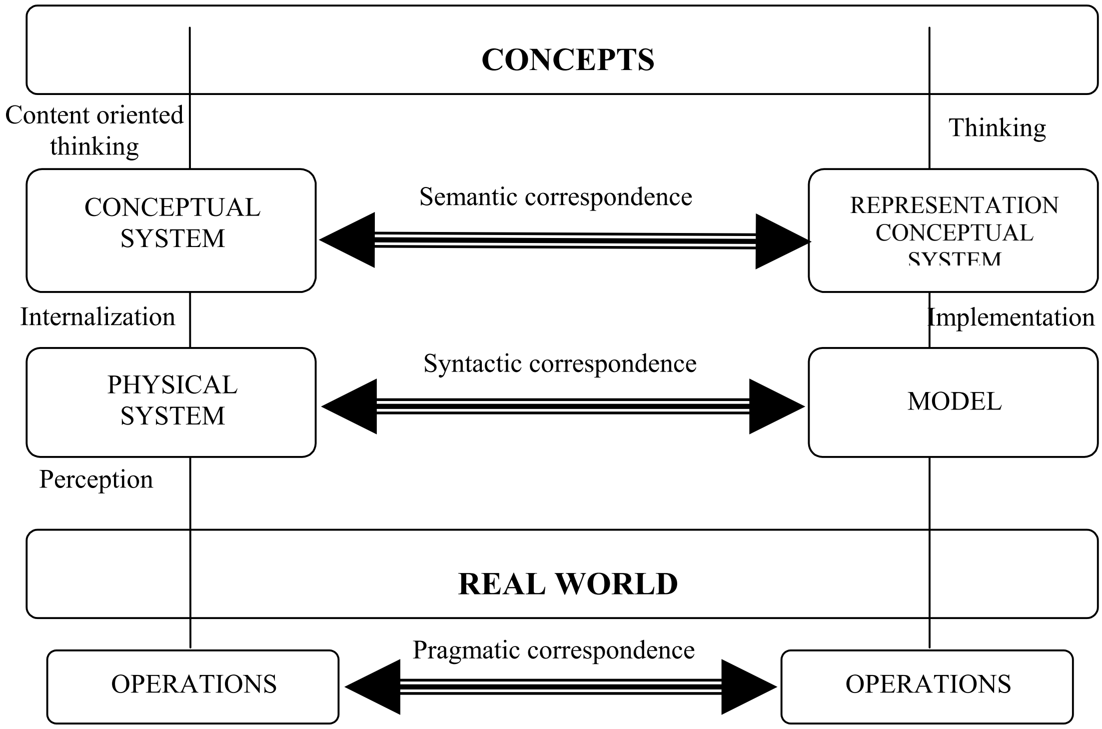


Figure 1.1: Correspondences between real world and models.

It is a usual phenomenon to blur the concepts of 'real world' and 'physical system', because people tend to think in their standard utilitarian perceptual interpretation. This kind of thinking may be often socially or professionally influenced or determined. The factors that influence the interpretation of real world are mentioned often as preconceptions in cognitive science. Every interpretation is basically arbitrary whatever its accepted usefulness and the accuracy level of the related sensors [Kirbach and Schmidt, 1976].

1.6 Epistemological value of models

After the above approach to the 'model' concept there are some questions that may naturally rise about the epistemological value of models. What is the relation of model with epistemological concepts like 'theory', 'law' etc. and what is the role of modelling in the scientific method.

There is no consensus among scientists on what is 'the scientific method' or whether it exists, and it is not the purpose of this document to argue either case. Having said this, 'the scientific method' is very often quoted and described as a four-phase process:

Phase 1.

During this phase scientists state the goal of study (problem to solve).

Phase 2.

The second phase involves design and implementation of certain experiments and observations. During the design of the experiments several concepts must be defined and their characteristics must be quantifiable in order to be measured. The selection and definition of the concepts related to the study goals and studied phenomenon/system is a no trivial creative and critical activity.

Phase 3.

The data collected during the second phase are used for the formulation of a model that resumes a large number of observations. The model behaves like the system under observation. The model represents the data and control structure of the observed system at a certain degree of accuracy.

Models do not explain the observed phenomenon but enable prediction of the behaviour in similar future observations. Every successful prediction consolidates the model's validity.

Phase 4.

This phase concerns the formulation of hypotheses for the interpretation of the collected data or the validation of the corresponding models. Systems of hypotheses for the interpretation of a set of scientific models concerning a specific field constitute a scientific theory.

A theory/hypothesis is never absolutely verifiable [Thornton, 1997]. A theory/hypothesis could be only falsified by counterexamples, i.e. by wrongly or non-predicted results of an experiment. The discovery of a theory's limitations is a first class opportunity for the progress of scientific knowledge. A new theory will eventually replace or complete the old theory.

Some times hypotheses are declared without experimental support (e.g. Avogadro's hypothesis, Democretos's hypothesis about atom). Many believe that hypotheses must be experimentally supported in order to constitute a scientific theory.

The above process is linearly described; however, this is not the way that scientists apply it. Scientific work is very complex and uses the intuitive power of scientists, which is not linearly describable.

1.7 Model evaluation criteria

As mentioned earlier for a physical system the construction of several models is often possible. These models are not equivalent. Models can be evaluated according to many even contradictory criteria. The following list contains some of the most important.

- Predictive power

Predictive power is the ability of a model to be executed and to produce behaviour proportional to the relevant physical system under analogous conditions. This enables scientists to predict the real system's behaviour without actual interaction with it. A weather model can be used in this manner to predict the future weather, or a scale model could predict the upper wind speed that a real bridge could afford etc..

- Explanatory power

This is the potential of a model to estimate the conditions that produce a certain observed behaviour. Explanation seems the reverse procedure of prediction. The model must be reversible in order to facilitate the explanation effort otherwise explanation becomes a trial and error procedure.

- Parsimony

A simpler model with comparable the other axiological characteristics is preferable than a complex model. Parsimony makes a model more understandable and engaging.

- Cost and Complexity

Cost is a self-evident evaluation criterion. Because of the increasing percentage of the computational models, cost often means computational complexity rather than economic cost. A 100% accurate weather forecasting model is useless if tomorrow's weather prediction needs 3 days of computational time.

1.8 The modelling process

The modelling process is iterative and adaptive. It includes a dynamic, two-way process of mapping, translation, transfer, and correspondence between the structural elements of the representation and the conceptual network of the field under study. This process will be called conceptual re-framing and will be considered as an important learning mechanism during modelling. The prediction of the existence of unknown planets while exploring the relationship between planetary model behaviour and the sky could be considered as a historic example of successful conceptual re-framing.

The adaptability of a model originates from the comparison of the model with the physical system in order to determine if the entailment is acceptable according to the study goals and the given resources.

1.9 Representation systems of models

A model needs a representation system in order to be expressed and externalised. There are several representation systems (often for the same problem domain) with different levels of formality and abstraction. A traditional model representation system is language. Language is very abstract and consequently very expressive. The price for expressiveness is ambiguity. Scientists are developing artificial inscription systems (representation systems) using natural language and other graphical symbolic systems in order to express scientific knowledge descriptions in a more economic and explicit manner (unambiguously). Representation systems are evolving with time and many believe that the introduction and adoption of a new one constitutes fundamental scientific progress. The introduction of Descartes's analytical geometry could be considered as a historical example of the impact of a new representation system in the progress of science.

1.9.1 The structure of representation systems

Scientific representation systems are structured like language and any other inscription system. Representation systems have an alphabet (basic symbol set), syntactic rules, semantic rules, and pragmatics. Syntactic rules describe how basic symbols could be combined in order to produce more complex and syntactically correct sentences. Semantic rules are used to interpret correctly the syntactic constructs of the language in another meaningful system.

The structure (alphabet, syntactic rules, and semantic rules) knowledge is not enough in order to use efficiently any non-trivial inscription system. There are idioms and verbiages (patterns of use).

1.9.2 Representation systems and problem solving

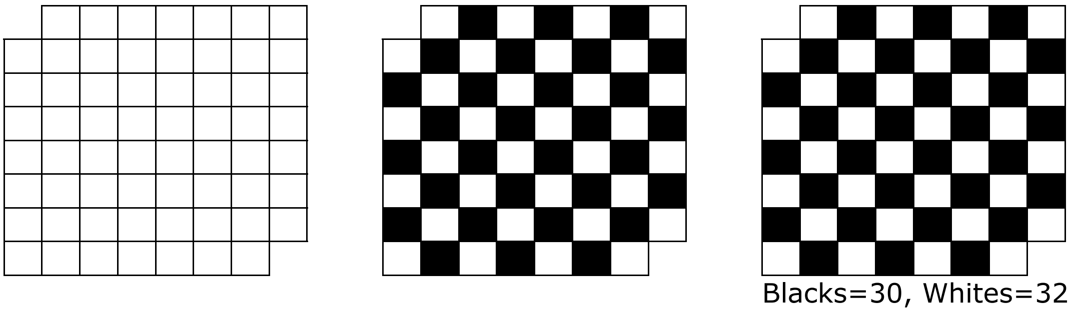
> *Even for human problem solvers a representation shift may make an enormous difference in problem-solving effectiveness.*
>
> *[Rich and Knight, 1991]*

Representation systems do not only amplify mental capabilities (memory and thinking power) but also reorganize mental functioning [Pea, 1985]. It is known that the choice of the representation system determines the solubility of several problems. A well-known example can be presented using the mutilated checkerboard problem, which can be stated as follows:

The Mutilated Checkerboard Problem

"Consider a normal checkerboard from which two squares, in opposite corners, have been removed. The task is to cover all remaining squares exactly with dominoes, each of which covers two squares. No overlapping, either of dominoes on top of each other or of dominoes over the boundary of the mutilated board is allowed. Can this task be done?"

One first idea to solve this problem is to try to enumerate, exhaustively, all possible tiling. Consider the three ways in which the mutilated checkerboard could be represented in the next figure:



Representation1: Uncoloured checkerboard.

Representation2: Coloured checkerboard.

Representation3: Coloured checkerboard annotated with the fact that black squares are less than white.

Figure 1.2: Multiple representations for the mutilated checkerboard

Representation1 does not suggest directly a problem solution. Representation2 is more helpful but the problem is trivial using representation3.

The familiarization with several representation systems is a strategic advantage for a problem solver.

Even the transformation between two representation systems of the same problem may be useful for gaining a better understanding. Modelling is a first class opportunity to learn about different representation systems and develop a rich representation experience.

1.9.3 Representation system evaluation criteria

It is possible to name several axiological criteria for representation systems; some of them are:

- Formality

The formality of a representation system is determined by the explicit existence of syntactic rules. For example, programming languages are of high formality in contrast to concept maps or other drawings. Formality enables the execution of the corresponding models but constrains the expression power of the representation system and usually makes longer the learning curve of the related tools.

- Abstraction level - Generality

A very abstract representation system like the one of mathematics logic gives expression power and wider application opportunities to the modeller but it makes also the representation system more difficult to familiarise with and use effectively.

- Accessibility

The representation system determines the number of people that are able to understand the produced models. Usually formal representation systems are understandable by fewer people. Special interest appears in representation systems that could be used by scientists and by non-science people.

- Thinking, amplifying and reorganizing

Representation systems provide conceptual frameworks that make several physical systems meaningful. Representation systems amplify and reorganize human thinking [Pea, 1985]. The degree of the impact of a representation system on human thinking is an interesting axiological criterion, as far as it is possible to estimate this impact.

- Conceptual framework suitability

The suitability of a representation system's conceptual framework for a systems domain is determined by the convenience and guidance that it offers for the establishment of correspondence with the physical system's conceptual network.

- Learning cost and convenience

Different representation systems require different time and effort for familiarization. Some systems require the existence of computer tools in order to be manageable and other require paper and pencil.

1.10 Categories of models

It is possible to categorize a model according to several criteria. Some criteria are general and other are based on the characteristics of the representation system.

1.10.1 Abstract vs. physical

A model could be material or abstract. Well-known material models are the physical scale models (replicas) used by engineers to understand and design complex systems such as bridges, houses, airplanes etc. Abstract models usually are expressed in a manner that does not indicate the physical structure of a system. (E.g. An equation system of a weather model does not have obvious similarity with the weather phenomena that it describes).

1.10.2 Formal vs. informal

Formal models are produced by formal representation systems in contrast with informal that are expressed using informal representations.

1.10.3 Procedural vs. declarative

Procedural models express explicitly sequences of actions that must be performed in order to model a process. A well-known procedural representation system is algorithms. In declarative models the modeller express requirements about the physical system and does not describe any procedure to obtain the requirements. The requirement's procedure is implicit in the representation system. For example in the language Prolog the user defines facts, and rules and the system applies the resolution procedure to answer all user-specified questions. One ambitious idea behind declarative modelling is that in such environments the user declares the 'what' and the system can find the 'how' - probably in an optimal manner.

1.10.4 Quantitative (numerical) vs. qualitative

In quantitative models the interesting characteristics of the physical system are quantified and expressed, as numbers while the relations between the characteristics are mathematical. In contrast in qualitative models the characteristics of the physical system as well as the relations among them are represented by abstract symbols (often graphical). Qualitative models can be informal such as concept maps or formal like semantic networks of knowledge representation.

1.10.5 Parallel vs. sequential computational models

Parallel models include actions that take place simultaneously by different agents. Sequential models exploit only one action-processing agent. Parallel models expressed in proper representation systems like data-flow diagrams, petri-nets etc. simulate parallel systems naturally.

1.10.6 Centralized vs. Distributed

In centralized models there is an agent that synchronizes or coordinates all actions (e.g. the control unit of a processor). In distributed models there are many coordinating agents that are interacting to produce the overall behaviour.

1.10.7 Dynamic (executable) vs. static

A model that is (not) producing any kind of behaviour is considered as dynamic (static) model. A dynamic model can be controlled to produce part of the real system behaviour enabling predictions (*'what if'* scenarios exploration). Furthermore if a model is reversible could be used to estimate the initial conditions that produced a specific behaviour, thus enabling explanations.

A dynamic model is possible to be executed. Execution means, to take a sequence of states corresponding to a sequence of physical system's states. When a dynamic model is executed, it simulates the real system. Simulation serves to solve problems of knowledge about systems.

1.11 The impact of computerized modelling on science

1.11.1 Modelling and Computing

Computers amplify the power of traditional models by accelerating the computations behind them. Many Algebraic and calculus models could not easily be applied in non-trivial cases because of the computational work required. Operations' research is another discipline in which models' applicability has been expanded by the computational implementations of the corresponding algorithms.

The impact of the accessibility of cheap, accurate, and quick computation can be realized in activities that productivity tools like spreadsheets and database management systems are used. Another way is to compare the computation of a logarithms' or trigonometric table by hand and by a computer program. Using computational media it is possible to free problem solving from the tedious algorithmic arithmetic, thus saving time for more interesting thinking.

Proper computational environments reorganize cognitive functions [Pea, 1985] offering new conceptual frameworks for modelling. Formulations like trees, graphs (networks), flow networks, logic, fuzzy logic, cell automata, autonomous agents, evolutionary models, object/event, entity relationship models etc. are giving new opportunities for modelling activities.

Computer models facilitate the study of phenomena:

- very fast or very slow (e.g. fire, species evolution)
- that take place in very small or very large space scale (e.g. nuclear models or integrated circuits, solar system),

- dangerous, impossible or expensive for real reproduction or interaction (volcano studies, seismology, nuclear experimentation, weather change environmental effects etc.).

1.11.2 Existing computerized modelling tools used in science

Scientists use extensively two kinds of modelling software, the first concerns general-purpose modelling environments and the second specialized modelling and simulation software. The detailed analysis of special purpose modelling software is very difficult task because of the diversity of the disciplines and the big number of the available tools. This situation becomes clear by doing a search on computational cosmology or computational chemistry etc. in any of the popular search engines of the web.

General purpose modelling software, on the other hand is of special interest to ModellingSpace because these tools are used by scientists of different disciplines and thus can be used as a communication code between sciences. General-purpose scientific tools are of special interest for identifying the user requirements of ModellingSpace because they probably concentrate the general characteristics of modelling environments, which are useful in many disciplines. General-purpose scientific modelling environments could be categorized in the following main categories:

Mathematics software

Systems like Matlab [Matlab web site], Mathematica [Mathematica web site], MathCAD [Mathcad web site] etc. These environments have many mathematical methods and functions implemented and ready to use. Furthermore mathematics software usually includes a general purpose programming language in order to be extensible and/or to have interfaces to general purpose programming languages.

General purpose productivity tools

Software products like spreadsheets, database management systems, graphing software, data visualization software etc. These tools are used mainly as productivity tools in business but they are often used in scientific modelling. This category of software has special interest for ModellingSpace because the do not require extensive familiarization with specific mathematical or other formulations.

Except for scientific modelling tools, engineers use software like CAD/CAM, VLSI design software, 3D Solid modelling software etc. for design and education purposes.

1.12 Summary

The above sections discuss the place of models and modelling in sciences. They demonstrate how fundamental is the notion of 'model' for a number of disciplines, and how modelling itself is one of the core activities of scientific practice.

Summing up the important points made:

- In different scientific contexts, the word 'model' has varied meanings, but on the whole tends to be restricted in its uses to 'idealised representations' which attempt to explain why aspects of the real world behave as they do. In this sense, models take up an intermediate position between reality and theory.
- The modelling process is iterative and adaptive and requires conceptual re-framing of the subject under study.
- Factors to be considered in evaluating a model are amongst others: its predictive or explanatory power, its parsimony and its cost.
- Models can be expressed by a variety of representation systems, the choice of which may determine the solubility of a problem. Therefore, familiarisation with several representation systems is a strategic advantage for a problem solver.

- Criteria for choosing a representation system for a model include: its formality, its abstraction level (or generality); its conceptual framework; its accessibility by a specific audience; its suitability; and its learning cost and convenience.
- There are different kinds of models, but there is no unique way of categorising them. Some of the suggested criteria are based on the characteristics of the model's representation system.
- Computers have amplified the power of traditional models, but have also provided new representation systems and conceptual frameworks for modelling.

Finally, efficient employment and management of modelling tools are key capabilities not only for the young science student, but more generally for the future citizen.

2. Models and Modelling in Education

2.1 Introduction

As we have shown in the previous sections an essential part of scientific activity is the creation of models for the phenomena and situations studied. A number of these models end up being taught as part of the teaching of science in schools. It is therefore important to introduce our pupils to the modelling process, if we want them to be able to perceive the nature of these models and of scientific knowledge in general.

Modelling is also the essential component of human activity, since it is always present when an individual tries to understand and interpret the phenomena around him/her and make relevant predictions. Human beings perceive the world and make models of it. They can also produce these models in a discourse, in other words produce symbolic representations and linguistic expressions in order to transmit them to others. On the other hand, the individual who decodes these linguistic expressions creates a model which resembles the worldview that the original transmitter had and wanted to communicate (Johnson-Laird, 1994).

Martinand, Chomat, Larcher, Lemeignan, Meheut and Weil-Barais (1991) in their report on the teaching of physical sciences in secondary schools in France suggested the modification of the then teaching content so that it included modelling activities in which pupils would use, but also create models. In the following sections, we will try to support this suggestion by referring to the cognitive operations which are stimulated by a modelling activity.

We will then focus on computer-based modelling and we will examine its impact on learning and more specifically on reasoning.

Finally, we will conclude with an examination of the place/status of modelling in European curricula and mainly in the curricula of the ModellingSpace partner countries.

2.2 Modelling as a Cognitive Activity

The cognitive functions of models are various (to represent, explain, communicate, convince, anticipate, conceive, repair, control, etc.). Similarly, the cognitive processes implied in the elaboration and the utilisation of models are multiple (conceptualisation, formalisation, reasoning, etc.). However, there are not many researches concerning this subject in cognitive psychology. Despite the considerable number of works on problem solving, the researchers are little concerned by the modelling process of situations. The studies have essentially been concerned with the selective and structuring process of data collection, the planning process (top-down and bottom-up), different solving strategies and reasoning forms. The importance granted by many authors to the analogical reasoning, as form of preferential reasoning of problem solving, led them to minimise the importance of the conceptualisation processes. But modelling, conceived as a representational process, implies an important work of conceptualisation[1]. This conceptualisation does not concern only individual cognitive activities, because scientific concepts are social knowledge. Consequently, modelling implies necessarily a personal relationship with the social norms and the share of knowledge[2]. This explains the

[1] In a study concerning the inclined plane, Zacharoula Smyrnaiou (2000) has found a marked evolution between scholastic levels. The youngest pupils do not conceptualize the board used in angle and height terms as compared to the plane.

[2] We have thus been able to observe pupils of high school that assimilated modelling to the mathematical formalisation and that, consequently, refused qualitative models (in the area of the energy teaching where we proposed a model of the energy chain), Lemeignan and Weil-Barais, 1993.

diversity of the theoretical psychological frameworks used in the studies on modelling: (1) information processing theory and constructivism, (2) social interactionism. The first frameworks consider the individual confronted with situations and with systems of representation (natural language, diagrams, logical and mathematical formalisms, etc.). The second framework considers the social relations and inter-psychic functionings.

The following section reports on Gérard Vergnaud's works. He is one of the rare contemporary psychologists who, in the 1970s, drew the researchers' attention to the importance of conceptualisation and representation processes in the child's intellectual development (Vergnaud, Halbwachs and Rouchier, 1978).

Following this, we present studies conducted in the framework of a thematic research action concerning the teaching and learning of modelling piloted by Jean Louis Martinand (Martinand *et al.*, 1991).

2.2.1 Contribution of the representation theory proposed by Gerard Vergnaud

In the continuation of Piaget's works on psycho-genesis of knowledge (1946, 1974; Piaget and Inhelder, 1955) and taking up Vygotski's thesis concerning the role of semiotic mediations (1985a, 1985b), Gérard Vergnaud (1987, 1990, 1994) has proposed a general theoretical framework which emphasises the relationships that the subject has to construct in order to be able to understand and interpret situations, to communicate their purpose, to make predictions, inferences, etc. (cf. Figure 2.1).

The schema proposed by Vergnaud (1987) in a constructivist approach emphasises the role of the subject's actions and cognitive resources in the elaboration of knowledge. Piaget supposes indeed that the individual builds his/her knowledge in the unceasing interaction thread with objects: internal regulation process would assure a best adaptation of the individual to his/her environment. The assimilation process allows to assimilate new knowledge to those already onsite in the cognitive system and the adaptation process transforms cognitive activities so as to adapt to the new situations. Vergnaud (1987) distinguishes three functioning registers:

- the register of actions on objects (the knowing subject is dependent of the reality: the subject acts, manipulates and thus provokes changes and transformations in the world of objects);
- the register of mental representations (presented in Vergnaud's theory by the «invariants opératoires»[3], that is to say the "constant organisation of the activity associated to classes of problems");
- the register of symbolic representations.

In Vergnaud's theory, mental representations are cognitive constructions that the subject elaborates to yield regularities that s/he abstracts from his/her activity. These mental representations guide the subjects' activities. Thus, mental representations have two origins: subject's action and cognitive structures. The individual uses one's knowledge of objects and phenomena to select and process the relevant information. Calculations and inferences are made based on this information; they lead to the elaboration of action rules. Action rules have an essential role since they allow to generate new conducts. In this sense, they have an adaptive role.

The schema proposed by Vergnaud can help the comprehension of the cognitive processes involved in modelling. Indeed, the author emphasises the homomorphism between the three functioning registers.

[3] In Vergnaud's theory, the «invariants opératoires» are generative cognitive structures and are not information stocked in long-term memory as, for example, the mental models of Johnson-Laird.

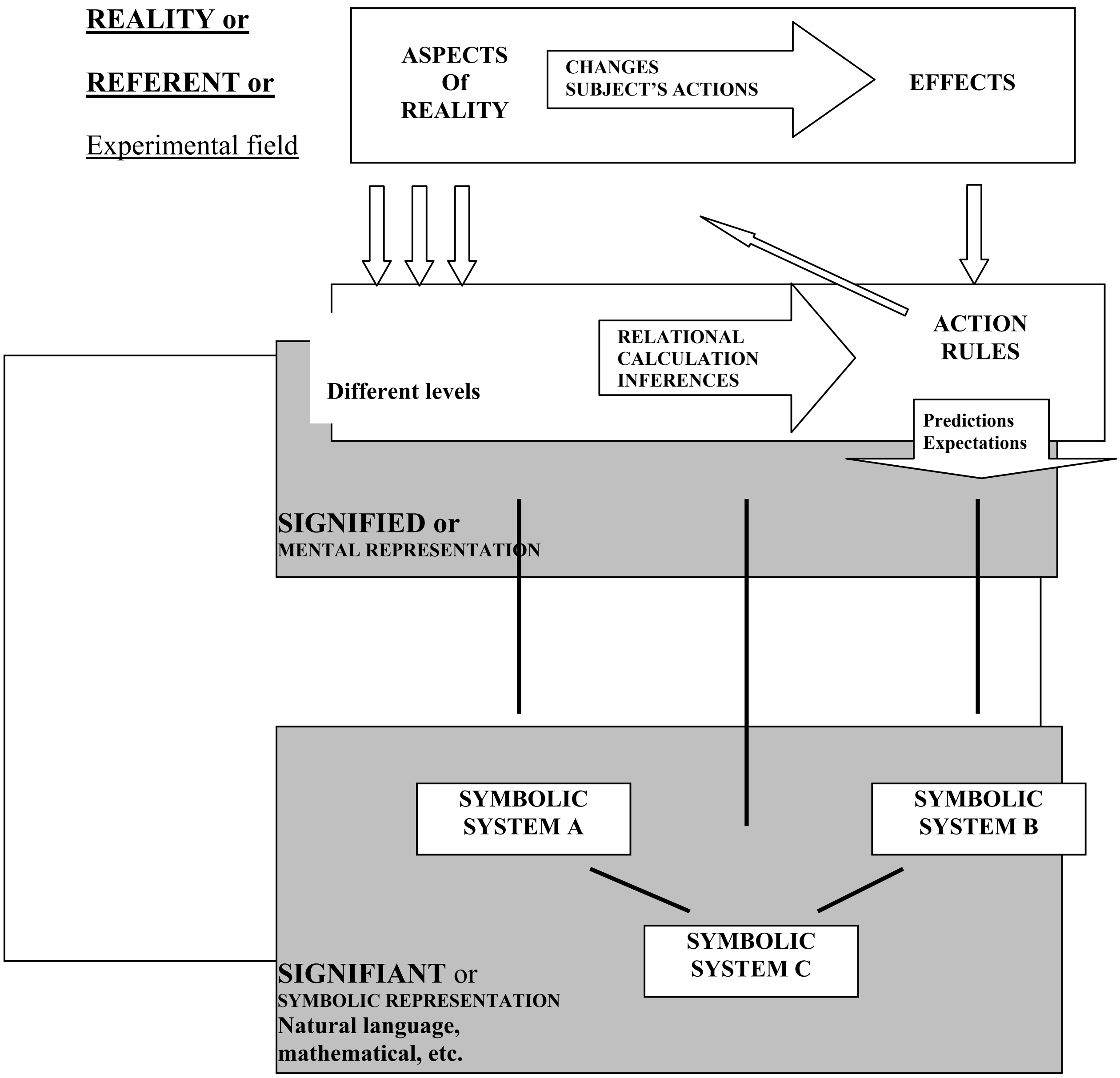


Figure 2.1: Concept representation, Vergnaud (1987)

Relationships between the three functioning registers indicate first that mental representations enrich linguistic and symbolic representations which can be used to represent concepts and newly acquired knowledge. Secondly, we can consider that games of translation between the different symbolic systems allow a return to the reality and enrich subject's knowledge of phenomenon. This is a very important aspect.

If one retains Gérard Vergnaud's idea, modelling mobilizes both perception and motor skills, mental representations and the «invariants opératoires» as well as the usage of semiotic systems. We point out that modelling is finalised by personal goals, which are objectives of action: to produce, to anticipate and to control effects. It results in the structures or the expression of relationships that allow inferences.

2.2.2 Modelling activity and reasoning

Modelling is often used for problem solving (Astolfi and Drouin, 1992), and thus demands different kinds of reasoning: deductive reasoning in which deductions are made on the basis of premises; inductive reasoning in which hypotheses are formulated from the observation of phenomena; and analogical reasoning. Weil-Barais (1993) defines reasoning both as the intellectual process itself and as the result of this process. This reasoning can be explicit (e.g. in the case of an argumentation), but in general is implicit and subconscious.

2.2.2.1 Scientific reasoning

Scientific reasoning consists largely of formulating hypotheses, planning experiments and analysing data from these experiments. It is characteristic of the formal operational stage of development, since it requires hypothetico-deductive reasoning, and thus is not present before adolescence. According to Piaget and Inhelder (1955), at the age of 11, the adolescent abandons the stage of concrete operations for this of formal operations. The adolescent is thus capable of making deductions on the basis of premises, and of formulating hypotheses on the basis of perceived (or non-perceived) realities. The hypothetico-deductive reasoning is rendered possible by two formal structures: the group of two kinds of reversibility (negation and reciprocity) and the combinatory. The combinatory operation is a classification of classifications and can be applied to objects but also to judgements and propositions. The experiments performed by Piaget and Inhelder concerning the equilibrium of weights, the oscillation of the pendulum and the floating of bodies demonstrate that the adolescents discover the basic principles by identifying variables and formulating hypotheses on the basis of their observations.

Johnson-Laird's (1983) approach to the development of reasoning is also very interesting, since according to this the adolescent, by treating the premises in the context of deductive reasoning, will create a mental model, that is a representation which will allow him/her to make inferences and reach a conclusion. This inferential activity, which consists of the use of a mental operation to arrive at a conclusion, is therefore essential for the acquisition of knowledge. For the conclusion to be valid, the individual must confront models which are alternative to his/her mental ones. We believe that what develops with age, are not the rules of inference, but the ability to look for alternative models. Hypothetico-deductive reasoning is not thus demarcated by a system of logic or by formal structures, but by the use of mental models. Moreover, Astolfi and Drouin (1992) specify that the search for alternative models averts the risk of confusing the studied reality with the hypothetic representation of this reality.

Induction consists of passing from observations to a more general law, by formulating hypotheses. This kind of reasoning is used in Piaget's experiments on the principles of equilibrium and of floatation of bodies. Even though the conclusions reached from this reasoning are not always valid, it is involved in a great number of scientific discoveries.

2.2.2.2 Analogical reasoning

Analogical reasoning is used by scientists, but also, more generally, by individuals in situations of problem solving.

The history of physics shows us that the search for analogies between two distinct categories of phenomena was very much used in the works of Kepler, Leibniz, etc.. From the writings of the 18th century on the subject of analogy, we can retain the critique suggested by Pierre Louis Moreau de Maupertuis: "The analogy rescues us from the pain of imagining new things; and from a pain even greater, which is to proceed in the uncertainty. Our spirit likes it: but does nature?" (free translation from French – cited by Hoffheimer, 1982). This quotation touches indirectly on the problematic relationship of model with reality. If science progresses using analogies to understand a new object of knowledge, this does not mean that the two domains which were brought closer in this way, are 'naturally' close. In general, it concerns the application of a well-known model of physics to the studied phenomenon.

Therefore, in 1955, Bar-Hillel expressed a certain contempt concerning the importance given to analogical reasoning. The author underlines that its "impact on the understanding of processes which are linked with the transmission of signals was extraordinary; it was so important that many people were made to believe that its success had to be imported to serve in other fields related in one way or other with 'communication'. Indeed, the analogical reasoning is certainly legitimate, but also often dangerous and misleading, specially when this does not accompany any attempt to discipline the thought".

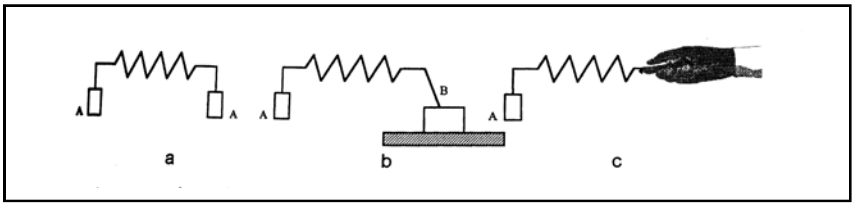
The psychologists are also interested in analogical reasoning. Miller (1983) defines analogy as the attempt "to explain something which is not well understood, by noting its similarity with something else which is well understood". The individual creates *analogical models* by using the properties of an object s/he knows, in order to explain the unknown properties of phenomena. A legendary example is this of the dark room as model of vision. Nothing demands that the analogical models are mechanical ones, but they often are. In modern times, the models are often mathematical algorithms, which however fulfil the same role. According to Weil-Barais (1993), even if the analogy is part of vague reasoning, the individual can nevertheless acquire new knowledge by constructing a representation of an unknown object on the basis of one s/he already knows. The analogy can also lead the subject to engage him/herself in scientific reasoning in order to validate the hypotheses s/he formulated concerning the unfamiliar object or phenomenon.

However, Duhem, after giving some examples of analogies, concludes that "these diverse ways of drawing analogies between two kinds of physics laws or between two distinct theories abandon in scientific discoveries, however they should not be confused with the use of models" (free translation of citation in Bachelard, 1979). Bachelard distinguishes the concept of analogy and the concept of model, by pointing out that although the two concepts are not totally unrelated, they do not have the same epistemological status. The analogical reasoning is thus considered as underlying any modelling activity, as a support to the construction of models.

These different kinds of reasoning will lead the adolescents in their construction of models, which will allow them to predict and understand various phenomena (Ferber, 1995, cited by Politis, Komis and Dimitrakopoulou, 2001). Just as the scientists construct models in order to understand better complex systems, the adolescents will create models in order to understand the natural phenomena. It is through the construction of these external representations that the individual will be able to elaborate on his/her internal representations (or mental models) of the phenomena (Papert, 1990, cited by Jackson *et al.*, 1995). By allowing a better understanding of these phenomena, modelling activities play a significant role in learning.

2.2.3 Teaching and learning through modelling (LIREST)

Studies concerning the teaching and learning of modelling conducted by the Interuniversity Laboratory of Research on Technical and Scientific Education (LIREST), from 1985 to 1995, focused on pupils' representations of physical phenomena (see for example Martinand *et al.*, 1991). These studies have established that when we ask pupils to make predictions concerning the evolution of a situation and to provide justifications, or when we ask them to explain a phenomenon, they are very able to formulate representations. However, in many areas, especially in mechanics studied by Gérard Lemeignan and Annick Weil-Barais (1993), pupils use essentially perceptual and motor information. Characteristics and functions of objects are supposed to play a causal role. For example, in the situation "spring" represented in Figure 2.2, pupils use the weights of the piece of lead to which an extremity of the spring is attached, to justify that the length of the spring will be longer in setting b that in setting a; or the fact that the piece of lead retains the spring to justify that the length of the spring will be shorter in b than in a, since if the piece of lead retains the spring it does not pull over. Comparatively to this information processing, where the pupil establishes a direct relationship between what s/he perceives and effects



a: objects (A) have the same mass.
b: a piece of lead (B) put on the table in which the mechanism allows the balance of the system.
c: a subject allows the balance of the system with the hand.

Figure 2.2: Experiment with spring, (Lemeignan and Weil-Barais, 1993).

expected, the modelling treatment depends on the usage of a relational invariant[4] (the relationship between the immobility and the sum of forces on the spring). This invariant (the model) is mobilised, according to experimental conditions (immobility of the spring, objects in interaction with the spring). That is schematised in Figure 2.3. How pupils can access such cognitive treatment? It was one of the aims of the studies conducted by researchers of LIREST.

Researches in LIREST have been concerned essentially with two knowledge areas in physical sciences: the mechanics at high school level, the matter at college level. In the first case, it is the Newton's model which is studied; it concerns a model expressed by quantitative relationships between magnitudes (Force, Energy, Quantity of Movement) (Lemeignan and Weil-Barais, 1987b, 1989). In the second case, it is a model of particles that is studied; it concerns a structural model expressed by drawings (products by pupils) and by a computer simulation of the movement of particles (used by pupils, in a second time).

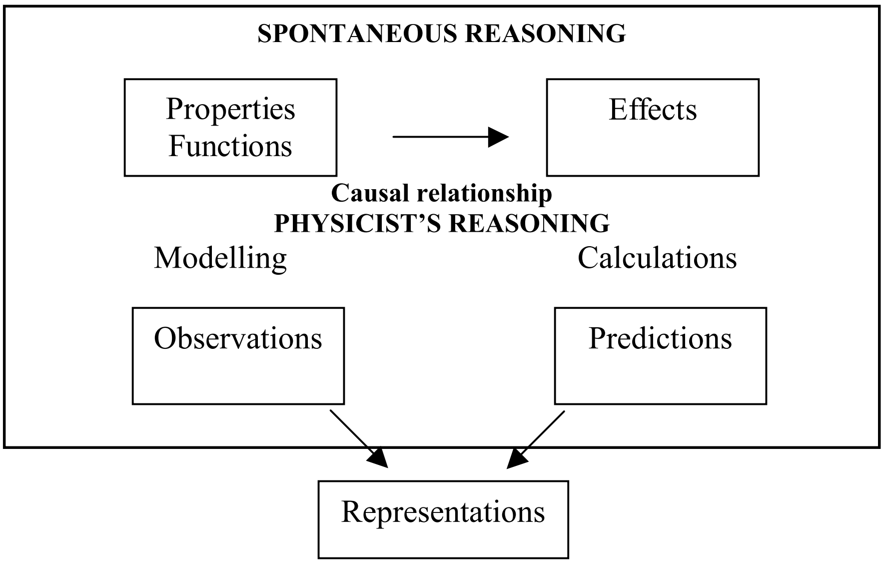


Figure 2.3: Construction of physical concepts (Lemeignan and Weil-Barais, 1993)

Considering the complexity of the studied scientific models, researchers invented "precursory models" susceptible to being constructed gradually by pupils. They are progressively constructed in reply to increasingly complex problems, due to the nature of phenomenon (displacement in a single direction or in the plane for example) or due to the diversification of the phenomenon (for example, expansion of a solid, a gas,

[4] In the example quoted, the model is expressed by a relationship between magnitudes. In others cases, this can be a structure (cf. part where models are described).

dissolution, distribution, etc.). These precursory models are constructed by reference to the concept of "Proximal Zone of Development" proposed by Vygotski (Weil-Barais, 1994a). The aim is to mobilise the cognitive resources of pupils and to bring them to construct models presenting similar functions to those of scientific models. These models are constructed to explain or to make predictions in limited experimental contexts. The extension of the experimental field explored by pupils leads them to grasp the limits of the model. In the same way, it justifies the transformation of the initial models.

Independently of the knowledge areas, the strategy employed consists of providing guidance for the construction of models. The relationships are first qualitative order (topological relationships, equivalence, order, implication, etc.). The construction of models instituting quantitative relationships between physical magnitudes comes only second. This choice is justified by preliminary studies which have shown pupils' difficulties to understand what describes physical magnitudes. This can be indeed, objects, systems (regrouped objects considered as an entity) or interactions between objects or between systems. Physical magnitudes can describe thus states or transformations. Precursory models that have been suggested by the researchers allow specifying these delicate distinctions at a cognitive level.

Several types of pedagogical process based on different cognitive activities have been tried out:

1. **constructive process** (a proposal being given, propose other compatible proposals with the first which allow to explain the new phenomenon);
2. **inductive process** (*research of causal factors* from the realisation of experiences, *causal relationship induction* from studies of co-variation and contra-variation, *quantitative relationship induction* from tables of measures (obtained from a software of simulation), *induction of a general relationship* from particular relationship discovered in particular experimental contexts);
3. **hypothetico-deductive process** (from hypotheses on properties or relationships, to deduce results and to confront them with experimental data).

These steps are more or less possible considering the status of proposals that form the basis of models (presumed or established in an empirical way). The pedagogical experiments have established that pupils adhere more easily to the inductive process and to the constructive process than to the hypothetico-deductive process (Lemeignan and Weil-Barais, 1987b, 1993, 1994). These experiments have confirmed the validity of proposals provided by researchers. Nevertheless, they emphasised the transformation of pedagogical practice necessitated by the modelling activities suggested. In fact, the apprenticeship of modelling presumes that teachers renounce practices in which teaching is seen as mere transmission of information. In the same way, teachers have to stop thinking of practical work as exercises of application and of experiments as supports for making pupils accept the relevant theory. In sum, the teaching of modelling asks for epistemological changes in teachers [Désautels and Larochelle (1992) make the same point]. So, it is necessary to change the training of the teachers before introducing changes in the scientific curricula.

The starting point of the modelling process is problems. The types of problems, the pedagogical settings, the tools provided to pupils and the tutoring[5]. determine the teaching that pupils receive.

[5] In order to emphasise the importance of teachers' practice in sequences of teaching which aim to teach pupils about modelling, Andrée Dumas Carré and Annick Weil-Barais ran a research program on tutelage and mediation (Dumas Carré and Weil-Barais, 1998). These researches pointed out the importance of language and interlocutory forms as well as the diversity of professors' intervention functions. They observed experienced professors who participated in the researches. The aim was to specify the interactions (between pupils and between pupils and the professor) that assist apprenticeships and not to describe usual class practice.

2.2.4 Summary

In the above sections we have established that modelling, seen as a representational process, implies an important work of conceptualisation, both at the individual and at the social level. This conceptualisation on the other hand implies a certain number of developmental constraints in relation to modelling activities. These concern the phenomenological domain accessible to children, concepts, mental models and cognitive skills that they are able to mobilise as well as semiotic systems that they are capable to use. These constraints are important. They concern particularly the physical dimensions that are constructed very slowly during cognitive development[6]. Anyway, there is an important time gap between the learning of formal systems and the possibility to use them[7]. One knows also that children and scientists' causal explanations of phenomena are different - and the same is true for adults (cf. works on commonsense representations: e.g. Black and Lucas, 1993; Driver, Guesne and Tiberghien, 1985; Giordan and De Vecchi, 1987; Joshua and Dupin, 1993).

More particularly, we have established that the formalisation of relationships is a general problem for pupils. Games of translations between relational expressions expressed schematically or in natural language and systems of formalisation (the logic and mathematics especially) are difficult, even for high school pupils (16 years). This legitimises the development of tools which allow pupils to get involved in these 'games' of translation with intent to understand the meanings in the different systems of representation. For us, this would have to be one of the essential cognitive contributions of ModellingSpace system.

Moreover, a good modelling tool should allow pupils to externalise and act on their thoughts and ideas, by providing appropriate and flexible primitives and structures. In turn, provided that these structures are well chosen, they can become tools for thought in other situations (Bliss, 1994a).

The last remark concerns the tools given to pupils (tools for representation, tools for simulation, tools for calculation). It appears that they acquire the status of cognitive tools only in problem solving situations, when the teacher gives to the modelling process an important role. Teacher's knowledge about the cognitive activities implicit in modelling as well as of the difficulties encountered by pupils are reference points for his/her interventions (Franceschelli and Weil-Barais, 1998; 1999). So pupils' activities are determined largely by the questions and situations provided by teachers and by the forms of tutelage and mediation that these teachers use. This is why we consider that innovative pedagogical tools have to be accompanied by tools for the training of teachers, if one wishes to achieve the training and apprenticeship objectives aimed for (Weil-Barais, 1997).

2.3 Computer-based modelling and reasoning

In the first section of this report we examined the impact of computer-based modelling on science, here we will examine its impact on education.

Alongside the more traditional models (e.g. physical or scale models, analogue models, etc.), computer models are increasingly being used in schools. These have enormous potential for allowing pupils to explore and interact with ideas in a way that was not possible before. And this is because they can provide "both new tools for thought and

[6] For example, the temperature is not conceived in the beginning as a measure. For the child, hot and cold are two distinct entities. It is only at the end of primary school that hot and cold are conceived as two modes of the same continuum.

[7] For example, we have been able to observe pupils of high school that they do not formalise relationships between measured magnitudes (mass, velocity) with mathematical relationships. For them, arithmetical operators have a computing function but not a function of representational relationship between measures.

new thoughts about the world, through interacting with external but artificial worlds" (Bliss, 1994a, p31).

Learning new ideas about the world, may involve clarifying ideas pupils already have, or may involve coming to terms with new ones. Pupils' own mental models are often unstable, not fully formulated, or even confused with one another. Thus, *"interacting with a model in a computer which is clearly structured and runnable provides at least the possibility of clarification and crystallization of ideas"* (Bliss, 1994a, p32).

Moreover, *"the vary way a model works, and the way it uses a certain set of primitives (whether variables, logical rules or rule-bound objects) to construct a representation of an aspect of reality are themselves potential tools for thought"* (Bliss, 1994a, p32). This fact points to the need for the modelling tool to have its primitives and structures as visible and transparent as possible, so that the pupils can easily internalise them and use them further to think with.

Finally, the computer provides the pupils with the remarkable experience to create a world, which can act and respond; a world which s/he can then watch evolve and change.

2.3.1 Reasoning with modelling tools

A useful distinction of modelling tools according to the kind of reasoning they endorse is:

- Quantitative: These tools are familiar from science and mathematics. They can mainly represent quantitative aspects of situations. Fundamental to work with these tools are the concept of variable (independent and dependent) and the nature of the algebraic relationship between the variables.
- Qualitative: These tools use qualitative rules or structures, as in decision-making games, expert systems or models of grammars.
- Semi-quantitative: These are comparatively newer kinds of modelling tools. They involve *"thinking about systems in terms of rough and ready sizes of things and directions of change"* (Mellar and Bliss, 1994).

In the following sections we will refer to some of the characteristics of reasoning with each of these kinds of modelling tools, as these are reported by research studies conducted with pupils in the context of the Tools for Exploratory Learning Programme[*]. The emphasis will be on the lessons we can draw for the design of such a tool.

2.3.1.1 Reasoning with a quantitative modelling tool

One of the studies conducted as part of the Tools for Exploratory Learning project looked at how pupils of 12 to 13 years use a prototype tool Q-MOD (which allows models to be built using simple algebraic relations between variables) to support quantitative reasoning. More specifically, the research focused on those aspects which seem to be most central to quantitative modelling: pupils' understanding of the nature of variables, and of the relationships among them (Boohan and Brosnan, 1994).

They describe a number of different ways in which pupils used the 'boxes' in the tool, intended to contain variables, to represent various kinds of entity (Boohan and Brosnan, 1994, pp62-64):

- Objects and events: Sometimes 'variables' were given names by pupils suggestive of objects or events, for example 'cheese', or 'snack'. Thinking in terms of objects or events seemed to help pupils estimate values and attach meaning to them.

[*] The 'Tools for Exploratory Learning' programme was run by J. Bliss, J. Ogborn, J. Briggs, D. Brough and H. Mellar between 1989 and 1992. The programme was funded within the ESRC's Initiative on Information Technology in Education.

- Numerals: Some pupils tended to regard any value merely as a number, ignoring its units. Consequently, calculations were described uniquely in terms of operations on numbers, without regard for the meaning of what is being done.
- Stored reminders: A few pupils used 'variables' with which no calculation was done, just as stored reminders of a quantity.
- Variables as constants: Some pupils created variables and built links, but never or rarely seemed to change the initial value of the variable, treating it thus as the single correct value of this variable.
- Alterable constants: The most frequent view of variables, was one of seeing them as 'alterable constants', that is of taking a constant value, which however can be altered to fit the situation given.
- Variables: Only very few pupils showed a more general understanding of idea of a variable, disassociated from a particular discrete object or event.

Concerning the relationships pupils identified between variables, Boohan and Brosnan (1994, p65) identified three main ways in which pupils thought about them:

- Processing mechanics: In this view, "relations are looked at as a set or sequence of operations on quantities".
- Real-world connections: In this view, relations and values in a model are related to plausible situations in reality.
- Semi-quantitative relations: In this view, quantitative relations are treated as a set of mutual influences.

<u>Consequences for the design and use of a quantitative modelling tool:</u>

- The tool should give pupils the opportunity to relate variables and values to objects, since in this way they are given meaning.
- The tool should remind pupils that each specified value for a variable has units and is not just a number.
- It should make obvious to the pupil that each dependent variable can have its value changed, and it should encourage pupils to manipulate these values in an easy, direct and continuous way.
- It should allow for the construction of abstract variables, which won't be associated with any particular object or event.
- It should allow for the construction of sequences of relations among the variables.
- It should allow for the creation of realistic models.
- Finally, it should provide pupils with the tools to approach the situation to be modelled both in a quantitative and semi-quantitative way, if they wish to.

2.3.1.2 Reasoning with a qualitative modelling tool

Work of the Tools for Exploratory Learning research programme also examined the nature of children's reasoning with a qualitative modelling tool. The focus was on two issues: the ease or difficulty that pupils experience in: a) moving from talking about a problem in a qualitative narrative way, to modelling it as interconnected choices; and b) perceiving a sequential model as a system of options (Bliss and Sakonidis, 1994).

It was found that pupils ranging in age from 11 to 14 found it not trivial to *"pass from the complexity and richness of the narrative to a simplification of the situation as represented in the model"* (p159). Furthermore, most of them also found it difficult to build paths in their models which had a sense of consequentiality (i.e. of relevance and continuity). Finally, when exploring already made models, the majority of pupils did not manage to see them as a set of options and did not manage to synthesize the consequences into wholes, so as to compare and evaluate them; their accounts rather remained at the level or re-descriptions of the contents of the models, with some elaborations.

In considering these results one has to take into account that the modelling qualitative tool used in the research included no processing power – it could not 'compute' consequences. As a result, the intended processing of information by the modeller could not be made explicit, nor was it supported by the system. This may have generated some of the difficulties that were reported in the study (Tompsett, 1994).

The above results point to the need of having a qualitative modelling tool with processing power, which by providing a more formal structure for the descriptions entered would help make their scope more explicit and meaningful.

2.3.1.3 Reasoning with a semi-quantitative modelling tool

The idea of semi-quantitative thinking has gained importance in the last two decades and has been the subject of study by researchers in the Artificial Intelligence field in which context it is known as 'qualitative reasoning' [de Kleer and Brown, 1981; 1983; Forbus, 1984; Kuipers, 1994]. The reason for this increased interest in it is that it is ubiquitous in natural everyday reasoning, since it identifies ordering of quantity (e.g. 'big' and 'small') but not magnitude, and involves using terms such as 'increasing' and 'decreasing' to describe changes (Ogborn and Miller, 1994, p37). "In *other words, this type of reasoning involves seeing how in a complex system the rough and ready size of something has an effect on the rough and ready size of something else, which may, in turn, affect other things and might in the end, feed back to affect the first quantity."* (Bliss, 1994c)

Computational methods to process semi-quantitative information directly have a complex symbolic formalism and require very powerful machines. They are therefore impractical to use in education, where the corresponding modelling tools internally generate quantitative models (i.e. use traditional numerical iterative computation methods) which somehow correspond to the semi-quantitative description of the model (e.g. see IQON).

Research related to semi-quantitative reasoning suggests that:

- it is a precursor to quantitative modelling (Ploetzner, Spada, Stumpf and Opwis, 1990 - cited in Bliss, 1994c, p122);
- it helps pupils in getting insights into complex situations and thus in learning difficult ideas (e.g. the function/derivative relationship) (Nemirovsky and Rubin, 1992a; 1992b – cited in Bliss, 1994c, p122);
- it is present even in quantitative tasks (Santos, 1992 - cited in Bliss, 1994c, p125);
- it is present when thinking about causally connected entities, and thus in developing or understanding causal diagrams (ibid);
- it seems to favour a diagrammatic representation as a natural means of expression (ibid);
- it allows pupils to approach system thinking from a fairly early age (ibid).

The above results provide positive indications for the design of computer tools to support semi-quantitative modelling and suggest their diagrammatic representation using a direct manipulation interface.

2.3.2 Computer Based Concept Mapping

A special reference will be made here to the role of computer based concept mapping in learning and reasoning.

2.3.2.1 The notion of concept mapping

Concept mapping is one of the most known ways for representing knowledge visually (Novak and Gowin, 1984; McAleese, 1998; Fisher *et al*, 2000) and consequently for externalizing the representations of those who learn by using graphs in a tree or network form (Jonassen and Marra, 1994; Fisher, 1990).

Concept mapping was developed by John Novak (Novak, 1977; 1990; 1998; Novak and Gowin, 1984) by elaborating the psychological views of Ausubel (Ausubel, 1968), who emphasizes the construction of links/associations between ideas. The basic ingredients of concept maps (see Figure 2.4) are the nodes, which represent concepts and the links, which represent relations between the concepts (Novak, 1977; Fisher, 1990; Buzan and Buzan, 1993; McAleese, 1998). Nodes that are connected with links form a concept map. Concept mapping is used in a similar way as the concepts of mind mapping (Buzan and Buzan, 1993) and semantic network (Quillian, 1968; Fisher, 1992). There are not essential differences among these three approaches (Fisher, 1990; McAleese, 1998) and, at least as far as education is concerned, tend to be used in a similar way. The procedure of concept mapping is conducted either by pencil and paper, or by using appropriate software, with the aim of helping students and teachers externalize their understanding of a concept and its relation to other concepts. In general terms, the software of concept mapping includes three basic elements: concepts, links and instances (Fisher, 1992; McAleese, 1998):

- A **concept** symbolizes an information unit and is represented by a word, a phrase or an image. In general, a concept is determined simply by its label, but it can also be represented by an image or animation.
- A **link** describes how a concept is linked to another. In general, a link is a relation that connects two concepts. A link can have a name or can simply show that two concepts are related one to another without further information.
- An **instance** expresses a sentence in the form of «concept – link - concept» and describes the relation between two concepts.

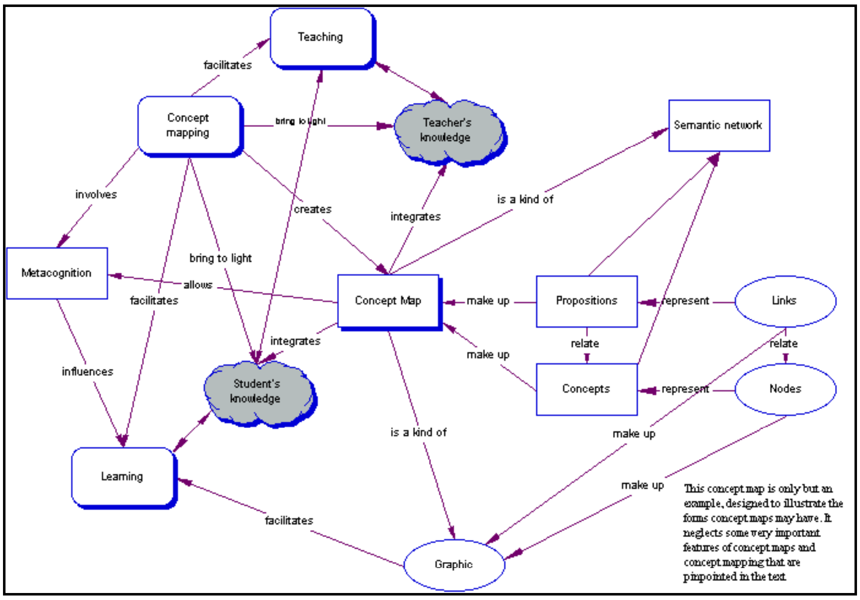


Figure 2.4: A concept map of Concept Maps (Baron *et al.*, 1999)

From a theoretical point of view, concept mapping can be a rather effective way of learning because it requires explication and reflection (making explicit what is normally implicit) and may help the pupil to develop *auto-monitoring techniques* and so to enhance their critical thinking. According to McAleese (1994) the process at the core of concept mapping is the auto-monitoring technique, for personal or group knowledge presentation or re-presentation. Ideas are 'created on the fly' and the learner has tools to use and operate through a series of stages (Figure 2.5).

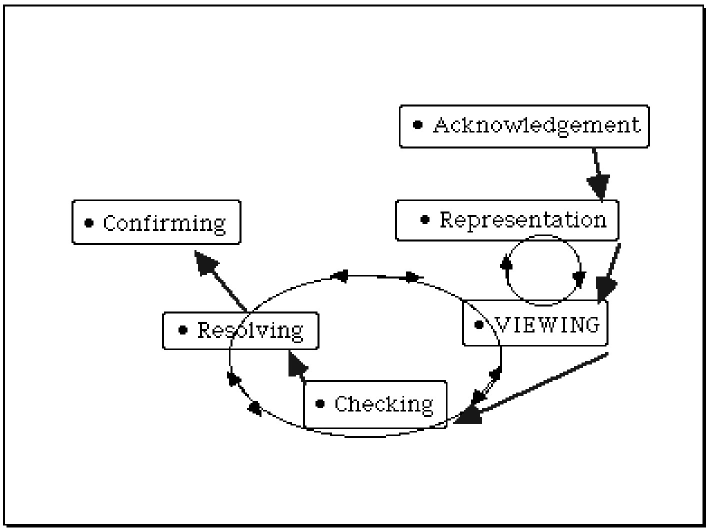


Figure 2.5: The Stages of Auto-monitoring (McAleese, 1994)

Many authors underline the importance of concept-formation tracking as a tool for monitoring pupils' conceptual growth over time (Anderson-Inman *et al.*, 1998). So, as a learning strategy, concept mapping is most effective if it is conducted on an ongoing basis over the course of instruction.

Concept mapping is used in various ways in the educational process (Novak and Gowin, 1984; McAleese, 1998; Fisher *et al*, 2000). Concept mapping facilitates both the teaching and/or the learning process and is used for evaluation purposes (i.e. by comparing two concept maps), for the externalization and record of the representations of a subject, for the exchange and communication of ideas, and for the design of multimedia applications and generally navigation systems.

2.3.2.2 The importance of Computer Based Concept Mapping environments

Over the past years a great number of software packages that supports the construction of concept maps has been developed (Anderson-Inman and Zeitz, 1993). A description and analysis of their technical characteristics and educational capabilities can be found in Baron *et al.* (1999) (see also Figure 2.4). From their analysis it is evident that the computational environment of most of this software does not allow collaboration among students. Only few allow communication in the form of map exchange in a synchronous or an asynchronous way *(CMap, Representation Tool 1.0, Belvedere, MindManager®, Inspiration® 6.0)* (see also section 4.4.1).

Anderson-Inman *et al.* (1998) observe that concept mapping is rarely used spontaneously by pupils, because it is difficult and that the process of map modification is messy and cumbersome. So, if we agree with the fact that modifying maps is essential because it allows learning to occur and conceptual understanding to grow, we have to offer learners more support in constructing networks and more encouragement to revise networks, so as to enhance their use in communication. Computer Based Concept Mapping (CBCM) environments can have a determinant role for that purpose, since revisions are much easier than on paper.

Anderson-Inman and Zeitz (1993) describe the benefits of concept mapping using computer software over traditional pencil and paper methods of organising information. They find that classroom use of Inspiration (one CBCM considered below) encourages users to revise or change their maps. They manipulate concepts and revise conceptual relationships. 'The practical advantages of constructing concept maps electronically are similar to those of using a word processing program to write. There is an ease of construction, an ease of revision, and the ability to customise maps in ways that are not possible when using paper and pencil' (Anderson-Inman *et al.*, 1998).

With CBCM, concept representations and their respective links are no longer static; both can be expanded as knowledge or elaboration of an idea increases. Errors in describing an idea can be easily corrected and adapted. Most computer assisted concept mapping tools allow the user to point and drag a concept or group of concepts to another place on the map and automatically update all the appropriate links (Anderson-Inman and Zeitz, 1993).

Another key point is that software usually allows the user to change his/her map to different electronic formats (e.g. from outline to graphic). These electronic formats can then be stored, sent, manipulated, used, printed, and deleted just like any computer file. Digital storage is especially important if concept maps will be re- used, completed by the author or others. It can facilitate co-operative tasks. Concept map in digital format can be easily sent as attached files with e-mail messages, or included in a WWW page.

Since concept mapping is a visual approach to organising our thoughts: (emerging relationships, thought patterns and themes) by visually clustering the idea symbols on screen, a common feature to all CBCM tools is that they allow the user to define objects (the nodes representing knowledge concepts) and to connect them with associative links that describe the relationships between them. But there are differences.

These differences may have their roots in a theory underlying the design of the software. For example, Decision Explorer is based on a 'Personal construct theory' due to George Kelly[8]. Mind Mapping is another well-documented approach invented by Tony Buzan (Buzan, 1995), following his research into note taking techniques. Comparing several techniques for taking notes during a lecture, he found that writing only key words lead to greater retention. With this result and other research, he suggested a new method for taking notes, using only key words and images, what he called mind maps (as brief and interesting for the eye as possible). These mind maps appear to be used in many different ways other than just taking notes. From these days, this initial idea has been developed and 'Mind map' is a registered trademark of the Buzan organisation.

It is a method for generating ideas by associations. In this theory, 'a mind map consists of a central word or concept, around the central word you draw the 5 or 10 main ideas that relate that word. You then take each of those child words and again draw the 5 to 10 ideas related to each of these words'[9]. In developing his technique of mind mapping; Buzan stated different laws (that can be summed up by several key words!): use emphasis, use association, be clear, develop a personal style, layout, use hierarchy, use numerical order. The products of mind mapping take these principles into account. Thus, an overview of MindManager describes how it fulfils the general laws given by Buzan[10].

2.4 Models and Modelling in European Curricula

As already mentioned computer based modelling in an educational context has attracted much attention in recent years, and there has been a rising interest in the development of educational modelling environments and corresponding curriculum materials for school use.

This section will examine in brief whether this interest is reflected in the curricula of the countries involved in the ModellingSpace project: Do they include references to modelling activities? Do they consider modelling among the concepts to be taught? The same questions will be discussed more extensively in the deliverable D05 (Scenarios and Activities).

Brief mention of the school curriculum of the UK is also included as a reference case, since it has a long history of teaching about modelling as part of its curriculum.

---

8 http://www.banxia.com/demindmap.html
[9] http://www.ozemail.com.au/~caveman/Creative/Mindmap/index.html
[10] http://www.mindman.com/english/comparison.html

2.4.1 The use of models in teaching

Before going on to discuss the use of the concepts of 'model' and 'modelling' in the curricula of the ModellingSpace partner countries, it would be useful to refer to a classification of these uses as this is reported by Gilbert and Osborne (1980; Osborne and Gilbert, 1980).

Gilbert and Osborne adopt the classification suggested originally by Black (1962). He classifies models into five types: scale, analogue, mathematical, theoretical and 'archetype'. Osborne and Gilbert (1980) choose the three first categories as being the ones more used in teaching.

- *Scale models*: These are *"likenesses of material objects, systems or processes, whether real or imaginary, that preserve relative proportions"* (Black, 1962). Examples are: models to show solar eclipses (in physics), parts of anatomy (in biology), Solvay Towers (in chemistry).

In Gilbert and Osborne's paper this class of models is taken to include diagrammatic representations of scale models and is consequently termed 'iconic'.

- *Analogue models:* These represent *"some material object, system, or process designed to produce as faithfully as possible in some new medium the structure or web of relationships in the original"* (Black, 1962). The analogue model *"shares with the original not a set of features or an identical proportionality of magnitudes but, more abstractly, the same structure or pattern of relationships"* (Black, 1962). Examples include: ball-and-stick structure of crystals (in chemistry), water wave behaviour of light (in physics), cybernetic representations of organ function (in biology).

- A *mathematical model* is one *"that can be summarised in, or represented by, a mathematical equation"* (Davies 1978, quoted in Osborne and Gilbert, 1980).

To these three categories we will add the category of 'theoretical models': these are seen as deriving from a specific theory, e.g. the particulate theory gives a model of the way electrons are arranged in atoms. These kinds of models appear also often in educational materials.

Clearly the above categorisation is not unique and the mentioned categories are not exclusive; they provide however a kind of 'compass' in our study of the use made of the term 'model' in the European curricula.

2.4.2 French speaking Belgian curricula in sciences and mathematics

The French speaking Belgian curriculum considers modelling as a basic competence.

This competence is explicitly mentioned in the curricula of both sciences and mathematics, as one to be taught in key stage 2, i.e. from year 3 of secondary school (13,5-14,5 year-old pupils) to year 6 (16,5-17,5 year-old pupils), and to be attested at the end of this key stage.

More particularly, in the curriculum of sciences modelling is considered as a transversal competence to be acquired both in the lessons of 'basic science' (which are addressed to all pupils) and in the lessons of 'general science' (addressed to science, mathematics or technology-oriented pupils). On the other hand, the use of computer based modelling is embraced only by the curriculum of 'general science'.

In the curriculum of mathematics, modelling is also considered as a competence relevant to all four thematic study areas of functions, algebra, geometry and trigonometry, and data processing, though perhaps more closely relevant to the study of functions. The use of modelling software is not a requirement; it is mentioned only once, as optional, in relation to the learning of geometry.

In the science and mathematics curricula of key stage 1, which covers year 1 of primary school (5,5-6,5 years-old pupils) to year 2 of secondary school (12,5-13,5 years-old pupils), modelling is not explicitly mentioned. However, the use of models is implicit in a number of other competences that pupils are required to master. For example, the science curriculum specifies that by the age of 11 pupils should be able to identify two variables and their values and express their relation (if applicable) in a quantitative way, or that they should be able to transfer acquired knowledge in other situations. Similarly the mathematics curriculum identifies as transversal competences at this stage, the ability to proceed to variations in order to analyse effects in problem solving and the ability to identify similarities and differences between properties and situations appearing in similar or different contexts.

2.4.3 French curricula in science and mathematics

Curriculum in mathematics

Pupils are initiated into modelling in lower secondary school: the objective is to learn about the scientific process. Starting from elementary school, they learn how to handle mathematical symbols and various types of representations (graph, drawing and diagram). The activities suggested in lower secondary school have as scope to enable pupils to establish relations between reality, (mathematical) symbolic systems and representations. These correspond to the three levels described by Vergnaud (c.f. section 2.2.1) and would be required at the time of modelling. The pupils learn the games of translation between the various systems of representation but it seems that there is no activity suggested which explicitly refers to modelling. In the field of data management, pupils must however use mathematical formalization to describe phenomena studied in other disciplines.

One of the objectives of the upper secondary school curriculum in mathematics is same as in lower secondary school to introduce pupils to modelling. More particularly, the curriculum distinguishes between the use of simulations, which are defined as "producing data with a pre-defined model" and the use of modelling, which is described as "associating a model with experimental data". However, whereas the use of simulations is prevalent in various topics (e.g. geometry), the activities of modelling appear only in relation to one quite specific field, this of probabilities.

Curriculum in science

The pupils start learning about modelling in their science lessons during lower secondary school: the objective is to familiarise pupils with the scientific method. From nursery school pupils begin to discover the world around them, to raise questions, to search for answers, to anticipate events and to explain them in words or using different sings/codes (the cognitive functions of the models: cf. section 2.2). From the real world, they gradually learn the scientific method: to observe, collect and compare data, etc. and the use of a specific language. For their work, they use various modes of communication and representation (written language, drawings, diagrams and graphs). The three levels: reality, mental representations and symbolic systems, described by Vergnaud, would be requested at the time of modelling (c.f. section 2.2.1). The term 'model' (model, scale models, diagrams, analogies or abstract structures) figure in certain works intended for the teachers and in the cards of activities that the teacher gives to the children. The functions of the model appear in an implicit way: they are used to describe (solar system), to explain and to predict (the pupils use the models to test their assumptions).

In the lower secondary school, in the official program, it is specified that the simple model of the atom, which is suggested, "does not claim to be a final representation of reality". The pupil must know that s/he will meet later more elaborate models of the atom. The models to be met later will be also more 'impressive' (which is to say that they make it possible to give an account of a greater number of experimental facts). In the school activities, one can meet the terms 'model' and 'modelling'. The pupils learn the theoretical (ideal) models, such as: the particulate model, the molecular model, the model of the luminous ray, the model of Rutherford, the model of Bohr and the current model of the atom.

In the upper secondary school, the official program provides a definition of modelling: modelling corresponds to "a development work of a simplified abstract representation of a phenomenon, which is required to identify the relevant parameters and those which are negligible in the situation given". The activity "can provide a qualitative comprehension of the phenomenon and possibly lead to an equation, the resolution of which will provide quantitative evaluations". The term 'model' is presented many times in the works of the pupils: in mechanics (modelling of the interactions...) in optics (model of the luminous ray...), in chemistry (the molecular model…), in the introduction to the temporal evolution of the systems (the modelling of the phenomenon of load (RC)), etc.

2.4.4 Portuguese curricula in science and mathematics

In Portugal mathematics is a subject of study for all students until grade 9 and for students of secondary schools oriented for courses in physical and biological sciences, engineering, architecture, etc. Mathematics is also offered for humanities courses in secondary schools, on an optional basis. Science (chemistry, biology, physics, geology) is also part of the curriculum for all students until grade 9 and for some courses on secondary schools.

Mathematics has undergone a significant change in the last 15 years, due to the influence of international curriculum movements inspired on the NCTM Standards (1989, 2000) and on research in mathematics education. Context-based problem solving, estimation, use of technology (graphical calculators and computers), hands-on experiments, exploration, visualization, etc. have increased in importance. On the other hand, algorithmic procedures, out-of-context problems, routine exercises, formal demonstrations, etc., have decreased in importance.

Science is mainly taught using lecturing and routine memorization. Curriculum documents establish experimentation and observation as compulsory strategies for teaching science but most students have little or very little opportunity to do real observations and experiments.

In 2002 a major reform is scheduled for basic (compulsory) and secondary education. It will reinforce more practical approaches to the sciences and mathematics. The following apply to the new curricula.

- In the 1st and 2nd cycles of basic (compulsory) school (i.e. from age 6 to 12), there is only one teacher for all subjects of study. Most teachers are unfamiliar with science topics. In the curricula documents, "modelling" is only used in the context of design and technology.
- In the 3rd cycle of basic (compulsory) school (i.e. from age 13 to 15) curricula, mathematical models are explicitly mentioned in the context of the study of functions, presented as "mathematical models of the real world". Focus should be given in particular to direct and inverse proportionality. Calculators and computers should be integrated in the curriculum. Software like spreadsheets, graphical tools, dynamic geometry tools, as well as internet environments, should be used in problem solving, investigative activities and projects.

  The mathematics curriculum gives more importance to modelling than the science curriculum, where modelling is not explicitly mentioned.
- At secondary school level (i.e. from age 16 to 18), same as at the previous level, modelling activities are only explicitly mentioned in the mathematics curriculum. Modelling is considered as an important way of doing mathematics, and is considered a transversal theme, inherent to all topics. Experimentation plays an important role on the curriculum, together with model building, using graphical calculators and computer software. The use of graphical calculators is compulsory, and students need to use them in the final grade 12 examination. Exploratory software, like spreadsheets, modelling software and dynamic geometry are also considered essential to the curriculum. 'Modellus', a software environment for modelling in

science and mathematics, developed in Portugal but also available and used in other countries, is reported as "unavoidable" in the curriculum. There are many modelling based activities available for teachers and students, either on paper or in electronic formats on the web.

- The secondary science curriculum is less explicit about the importance of modelling and in some cases considers mathematical modelling as something that must be avoided and substituted by more qualitative and descriptive approaches. Only in grade 12 mathematical modelling has a relevant place in physics, because mathematical models are used extensively, even thought the term "modelling" is not used in the curriculum documents.

2.4.5 Greek curricula in mathematics, ICT and science

In 2003, new curricula will be enforced for all school subjects of compulsory education (Years 1-9, ages 6-15) in Greece. To these curricula, and more specifically to those in maths, ICT and science, we will be referring in this section.

- In the new Mathematics curriculum the development of problem solving capability is considered as the central goal of maths education, around which the teaching of basic maths concepts should be organised. By acquiring this capability pupils are said to become capable amongst other things to apply the modelling process in the study of real-world situations. Additional goals are considered to be both the practical application and the design of mathematic models in the context of problem solving. The use of appropriate educational software from the early years of maths education is also thought as desirable, though supplementary to teaching. It is seen on the one hand as extending the limits of a given representation, and on the other, as offering the possibility of having multiple representations of a concept in conjunction with the evolution of a phenomenon or event.

How are these goals translated in the maths curricula of primary and lower secondary school? The primary maths curriculum includes only reference to the use of 'iconic' models for the study of fractions. In the lower secondary curriculum, the use of mathematical models (to represent and understand the relationships that link various quantities) is explicitly mentioned as one of the objectives for the teaching of Algebra.

- The new ICT curriculum stresses the use of computer in teaching, as knowledge exploration tool, and promotes the use of appropriate open and explorative learning software. One of the goals of computer use is seen to be:

"To facilitate and reinforce the development of modelling skills and problem solving techniques."

At nursery and primary level ICT is to be used as a tool for educational content and should not be taught as a separate subject. Appropriate programmes to be used include: simulations, open learning environments, educational applications of explorative character.

In lower secondary school, pupils are expected to use simple application programmes for (amongst other things) modelling, in order to convey their observations and thoughts and to present scientific information using drawings, graphs, tables, words and text. However, the teaching recommendations for each school year do not mention of any specific modelling activity to be used. On the other hand, they specify that in Year 8 (age 13-14) pupils should learn how to create and use a spreadsheet, and in Year 9 (age 14-15) how to design the solution of a simple problem and execute it in a programming environment. Relevant activities are suggested in both cases.

- Finally, the introduction to the new science curriculum specifies that school science activities should have as objectives to re-inforce: the collaborative character of learning; the development of deductive and inductive reasoning; the best application of new technologies as tools for learning and thinking; *the development of modelling skills and problem solving techniques*; the ability to use symbolic means of expression and exploration; the development of transversal skills and capabilities of methodological character, and the development of an environment of mutual respect. Educational science software to be used is expected to include simulation programmes for the study of factors that affect physical phenomena.

In addition, the science curriculum indicates that pupils should be assessed (amongst other things) on their ability to:

- extract data from tables, graphs, etc., in order to identify the quantitative changes observed in physical phenomena;
- express hypotheses and create mental models in order to interpret physical phenomena.

All these, that is the specified objectives for science activities, the suggested specifications for educational software and the assessment recommendations are seen as very relevant and compatible to the ModellingSpace learning environment's concept.

How are these translated in the curricula of particular science subjects? Amongst the suggested objectives for the teaching of Physics and Chemistry are that pupils learn how *"to build and use scientific 'archetypes/models' in order to describe, interpret and predict some physical phenomena and processes"*. The accomplishment of these objectives is said to be facilitated by the use of new technologies / pedagogic tools which allow pupils to collect, analyse, visualise, model and communicate data.

In more concrete terms, in the primary Physics and Chemistry curriculum there is reference to the construction of simple physical models (e.g. of molecules), whereas in the respective secondary curricula, there is reference to more abstract/theoretical models such as the atomic/particulate/microscopic model. At both levels, educational software should (according to the curriculum specifications) include simulations of experiments or phenomena, which are not easy to perform in the classroom or to observe in the immediate natural environment of the pupils, or should present models for the interpretation of microscopic or macroscopic phenomena.

2.4.6 UK curricula in ICT, science and mathematics

Ideas related to modelling have been an integral part of the National Curriculum for England and Wales since its inception in 1988. These ideas are of three kinds - for pupils:s

3. to understand the concept of a model as a partial representation of some real or imaginary situation

4. to use computer–based models as a didactic tool to gain greater understanding of concepts in (for example) science.

5. to be able to create, modify, use and understand a variety of computer-based models

Examples of the first of these are found in the National Curricula of many subjects – for example the use of maps in geography, and the 'solar' model of atomic structure in science.

Examples of the second kind of use (as didactic tools) are not included in the National Curriculum as statutory requirements, simply as indicative examples. Further, the great majority of exemplar models listed in the science National Curriculum are not computer-based, most being similar in nature to the use of visking tubing as a model of the digestive system.

Examples of the development of expertise in the use of computer-based modelling are confined mainly to the National Curriculum for the (subject) ICT, which in England and Wales is primarily concerned with the development of ICT *capability*, rather than of expertise in any particular software package – e.g. MS Office. The extracts from each of the four 'Key Stages' of the National Curriculum listed below, exemplify the way in which it sees this aspect of ICT capability developing:

Key Stage 1 (ages 5-7): "Pupils should be taught… to try things out and explore what happens in real and imaginary situations [for example, using and adventure game or simulation]."

Key Stage 2 (ages 7-11): "Pupils should be taught… to use simulations and explore models in order to answer 'What if…?' questions, to investigate and evaluate the effect of changing values and to identify patterns and relationships [for example, simulation software, spreadsheet models]."

Key Stage 3 (ages 11-14): "Pupils should be taught… how to use ICT to test predictions and discover patterns and relationships, by exploring, evaluating and developing models and changing their rules and values."

Key Stage 4 (ages 14-16): "Pupils should be taught… (to) apply, as appropriate, the concepts and techniques of ICT-based models, considering their advantages and limitations against other methods."

(In passing it should be stressed that although the National Curriculum mentions 'simulation software' it does not list *any* proprietary software by name – and the substantive exemplar activities it gives use generic tools such as Logo or spreadsheet software.)

To help ensure that the National Curriculum is effectively translated into classroom practice the government has developed from it non-statutory but highly influential schemes of work for Key Stages 1-3. In addition, from September 2002, it will be implementing a 'Key Stage 3 Strategy', which has the aim (*inter alia*) of ensuring that pupils are taught (and assessed in) all aspects of the ICT National Curriculum.

Example activities from the Schemes of Work for each of the Key Stage 1-3 are given below:

Key Stage 1: *Activity 'An introduction to modelling'*

In this activity children learn that a computer can be used to represent real or fantasy situations. They understand that the representation is not an exact replica of the original. They discuss the main differences and similarities between a representation and the original. They create their own representations of real or fantasy situations.

Key Stage 2: *Activity 'Exploring simulations'*

In this activity children begin to understand that computer simulations can represent real and imaginary situations. They learn how to explore simulations, explore options and to test their predictions. They evaluate simulations by comparing them with real situations and considering their usefulness.

Key Stage 3: *Activity 'Models rules and investigations'*

In this activity pupils learn how simple models are built by first investigating rules, then by seeing how rules can govern the behaviour of simple models…. Pupils discuss the ways in which the model could be presented in a spreadsheet, identifying the inputs, the rules (formulae) and the outputs. Pupils then work in groups to construct this model, revising cells, formulae and cell references. They test the effectiveness of the model by using sample data representing a number of scenarios.

2.4.7 Summary

To sum up, the possible roles of models in teaching as these arise from an examination of the school curricula are:

- *Models as subject content* (especially in science). For example, in science students should come to know the major scientific consensus models (e.g. Bohr's atomic model) as well as the scope and limitations of such models.

- *Models as visualization and didactic tools.* These could be either:
    a. Material models (e.g. molecule construction using balls)
    b. Software models (e.g. simulations, graphing programs etc.)

- *Models to learn about science*: Students learn about the nature of models and how to appreciate the role of models in the accreditation and dissemination of the products of scientific enquiry.

- *Models to learn about modelling*: Students should be able to create, express and test their own models:
    - for prediction, design, evaluation of experiments etc. (mainly in science)
    - for familiarization with general purpose representation systems and improvement of problem solving skills (mainly in mathematics and ICT).

3. Users provisions and requirements

3.1 Introduction

This section contains a review of the state of the art in the use and provision of computers and informatic tools in European schools, based on surveys and reports conducted and published by the European Commission. This review is seen as setting the current scene of ICT and internet usage in schools.

There are also references to reports compiled in the UK by a government-funded agency (BECTa), which provides advice and guidance to teachers and the developer/provider community, with the aim of encouraging an increase in the availability of high quality software. These reports bring together the views of developers, teachers, local authority advisers, subject associations and government representatives, and provide some interesting insights into user requirements for modelling software.

3.2 State of the art in the use and provision of informatic tools in European schools

3.2.1 Incorporation of ICT into European Education Systems

In the 2000/2001 Eurydice Annual Report 'Basic Indicators on the Incorporation of ICT into European Education Systems' (Eurydice, 2001) one can find information on national policies for education in information and communication technology (ICT) and indicators on the way this technology has been incorporated into European school education systems.

The indicators contained in this publication draw special attention to a number of aspects. The ones more relevant to our purposes are:

- Education policies are increasingly geared to the use of ICT

A national or official policy encouraging the use of information and communication technology in education is in operation in all European countries. The integration of ICT into school systems is becoming progressively more widespread.

Moreover, one or more national projects aimed at boosting the computerisation of schools but also the practical use of ICT (whether this relates to teacher training, development of the skills of pupils or the use of educational software) in compulsory and upper secondary education have been initiated in all European countries.

- The number of pupils per computer is lower in secondary education

Secondary schools are better equipped with computer facilities overall. The number of pupils per computer (with or without an Internet connection) is nearly always lower than in primary education. More specifically, the pupil/computer ratios in primary and secondary education in the ModellingSpace partner countries are:

	Belgium	**France**	**Greece**	**Portugal**	**Spain**	**EU**
Primary	11,0	14,1	29,4	17,0	11,2	**13,2**
Secondary	8,0	9,4	15,2	16,4	12,4	**8,6**

(In calculating the number of pupils per computer, only schools where computers are used for educational purposes are taken into account.)

- ICT as part of the primary school curriculum

Even at primary level, learning about ICT has now become an integral part of the minimum compulsory curriculum in many countries. However, in a few cases the inclusion of ICT is a recent development.

Concerning the ModellingSpace partner countries, the Eurydice report states that in:

Belgium (B fr): Since 1999, ICT has been planned for inclusion among the 'core skills' in education, in which the competence that pupils should acquire in the subject is clearly specified. The inclusion of ICT in courses is compulsory with effect from 2001.

Greece: ICT is not part of the curriculum but the Pedagogical Institute has encouraged its use in a pilot project involving 40 primary schools.

Portugal: ICT has been part of the curriculum in primary education since the adoption of the statutory order of 18 January 2001. It will be one of the horizontal skills targeted by primary education with effect from 2001/2002.

Another interesting result is that in the primary schools which have included ICT in their curriculum, the majority of teachers regularly use computers with their pupils during lessons. The most widespread approach is to use ICT as a tool for educational content and/or, in some cases, for carrying out projects, interdisciplinary or otherwise.

The objectives pursued by the teaching or the use of ICT at primary level are distinguished in the report into four major categories, namely the development of programming ability, the use of various software packages, information searches and communication via a network. It is not obvious in which of these categories would 'modelling' fit. However, the development of programming skills is the least widely encountered objective.

- ICT as part of the secondary school curriculum

At both levels of secondary education, the teaching and/or use of ICT is virtually the norm in all countries. However, its use with pupils is undoubtedly less frequent than in primary education. Having said this, we need also to note that disparities between countries are also less marked at secondary level.

Interesting for our purposes is the result that in almost all EU countries, curricula for lower and upper secondary education refer to communicating via a network as among the objectives to be pursued. The practice however is still not very widespread. From the reasons give by teachers to explain this, the least commonly cited are those related to lack of familiarity with the Internet on the part of pupils or the inability of teachers to use it. Yet in France, Greece and Portugal these reasons appear relatively more significant and cited by close to 20% of teachers and indeed in France by over a third.

At this level of education, in almost all European Union countries ICT is taught as a separate subject, but teachers still often rely on it as a tool for teaching other subjects.

Concerning the development of programming skills: this objective is included in a greater number of curricula at upper secondary level.

- Teachers using or teaching ICT

ICT specialists are most likely to be found at secondary level. They teach ICT as a subject in its own right, whereas teachers of other subjects make use of ICT as a tool.

The report also states that in the EU in 2000/2001, 71% and 60% of teachers at primary and general secondary levels respectively said they used ICT with their pupils on a fairly regular basis. This use was not found related to either any official relevant training or to the age of teachers.

3.2.2 Internet in schools

The following extract from the *e*Europe 2002 Benchmarking Report [COM, 2002/62] describes the EU schools' infrastructure and access to the Internet.

Connecting all schools to the Internet by the end of 2001 was an *e*Europe target. This was all but achieved in May 2001 when more than 80% of EU schools were on-line (whereas 94% were equipped with computers). However, being a student in a school connected to the Internet does not necessarily mean that one has access to the Internet. Neither does it imply that the Internet is being used for learning. In over 10% of schools connected, pupils did not have access as the Internet was being used for administrative rather than educational purposes.

Attention must therefore shift to better connections and wider educational use. The Internet alone is not enough. Schools must be equipped to allow convenient and appropriate use; Internet must be effectively integrated into curricula; and, teachers must be supported and trained to use the new tools efficiently. The key results of a Commission report on Internet usage in schools can be summarised as follows:

i) On average, there are 12 pupils per off-line computer and 25 pupils per computer connected to the Internet (Chart 3.1). Half of these computers are less than three years old. However, there are considerable differences between Member States.

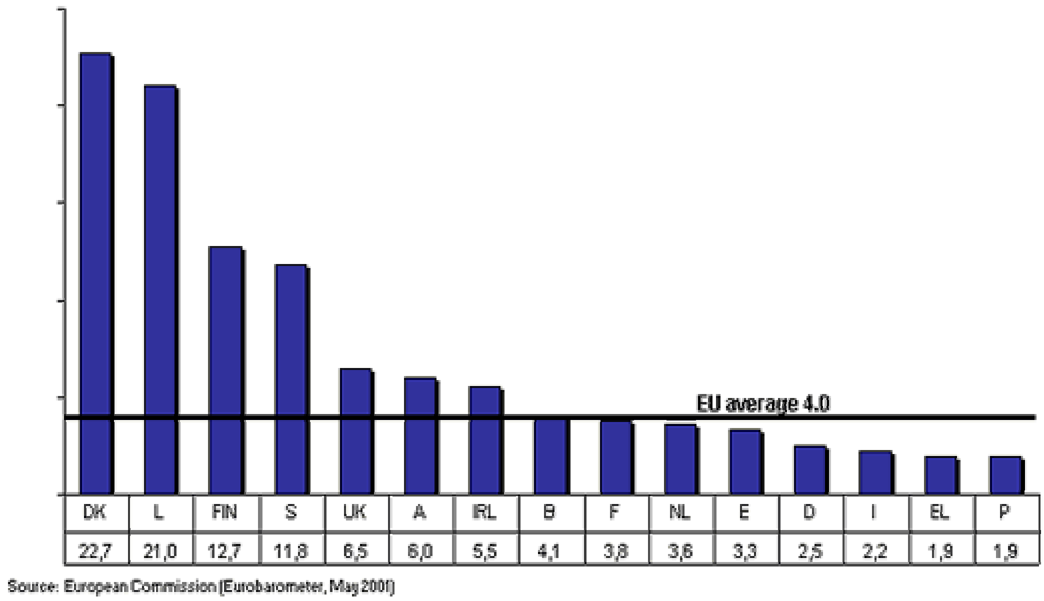


Chart 3.1: PCs connected to Internet per 100 pupils

ii) Connectivity remains dominated by narrowband technologies: over two thirds of school connections are ISDN and the others mostly dial-up via a regular phone line. Broadband technologies are marginal, although ADSL and cable modem are now more widely used in a few countries (Charts 3.2 and 3.3).

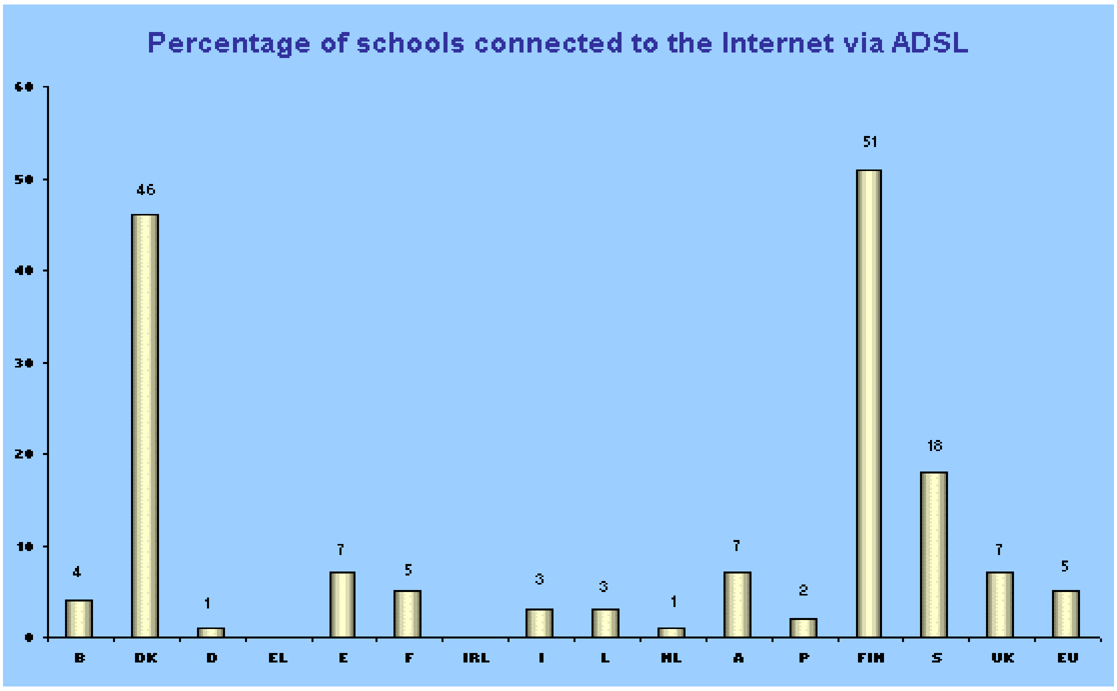


Chart 3.2: Percentage of schools connected to the Internet via ADSL (Source: European Commission – Flash Eurobarometer 101 and 102 - June 2001)

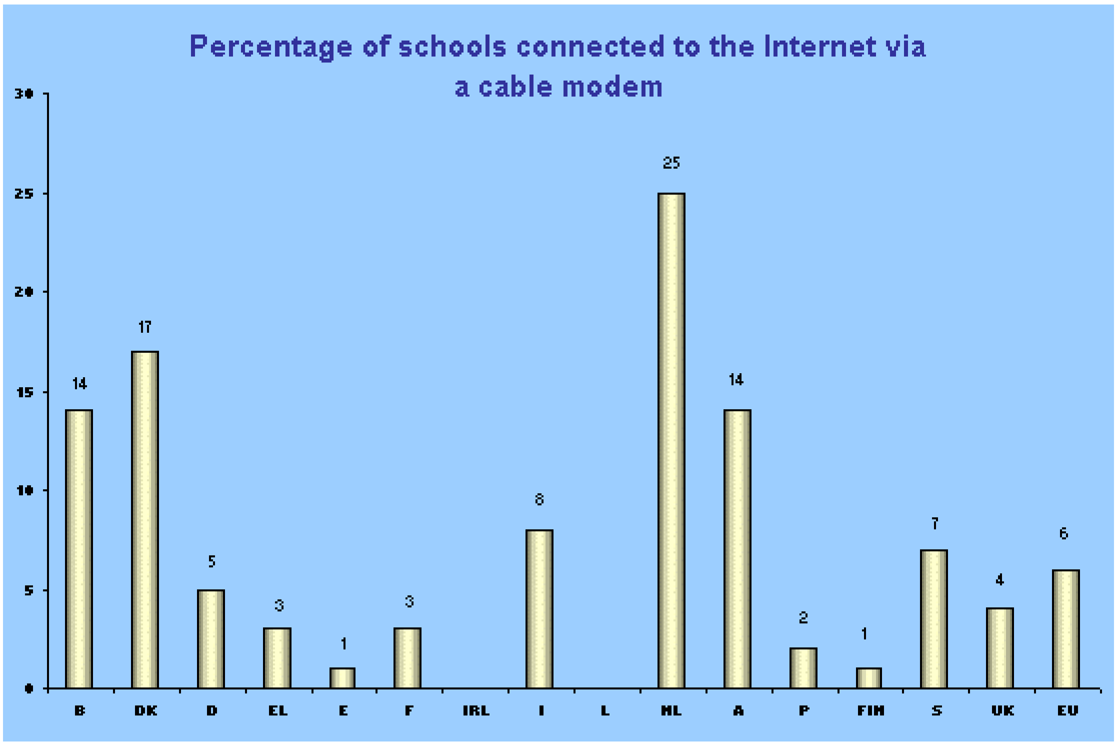


Chart 3.3: Percentage of schools connected to the Internet via a cable modem (Source: European Commission – Flash Eurobarometer 101 and 102 - June 2001)

iii) Whilst computers are now used by a majority of teachers, only a minority of them use the Internet for educational purposes (Chart 3.4). The main reasons given by teachers who do not use the Internet are poor levels of equipment and connectivity. Lack of familiarity does not seem to be a major problem. More than half of Europe's teachers have been trained in the use of computers and the

Internet, around 90% of teachers use a computer at home and approximately 70% have an Internet connection at home.

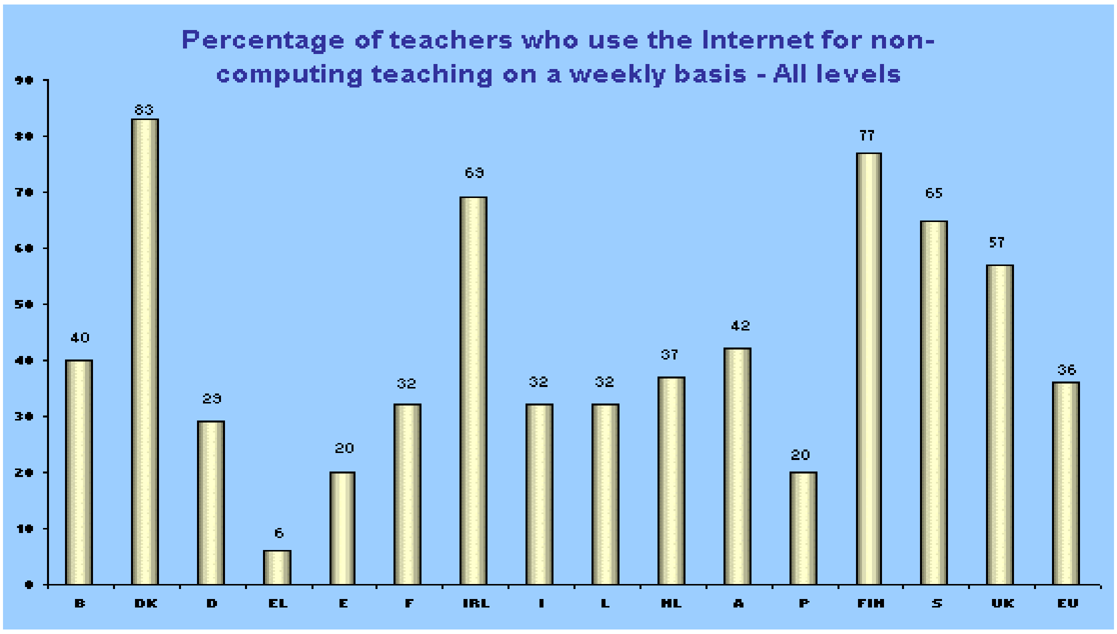


Chart 3.4: Percentage of teachers who use the internet for non-computing teaching on a weekly basis – All levels (Source: European Commission – Flash Eurobarometer 101 and 102 - June 2001)

In conclusion, there is a small group of pioneer countries that are ahead in terms of equipment, connectivity and usage. These Member States are benchmarks for the Union and worldwide. There are a small number of Member States that are lagging in almost all areas. In spite of this mixed picture, introducing Internet in education remains a priority in all Member States and European teachers seem to be open and well trained. Ultimately, all pupils should be digitally literate by the time they leave school.

3.3 Modelling as a User requirement

As mentioned before, the UK has been amongst the first countries to incorporate the teaching of ICT in the school curriculum. This is why we turned to them to see if they have published any information concerning user requirements for modelling software. Indeed BECTa (British Educational Communications and Technology agency) [website], a government-funded organisation with relevant objectives (see extracts below), as part of its Curriculum Software Initiative Project, has published a report for each curriculum area identifying amongst other things generic software requirements and content requirements.

These reports were compiled following seminars which took place during October and November 1999; they bring together the views of developers, teachers, local authority advisers, subject associations and government representatives, and provide some interesting insights into user requirements for modelling software. The following extracts refer to these requirements in relation to the curriculum subject areas of Geography, Science and ICT.

About BECTa/DfEE Curriculum Software Initiative

BECTa is a Government-funded agency in the UK working to ensure that ICT supports the government's efforts to drive up standards in curriculum subjects, in the teaching of key skills, in institutional effectiveness, and in the development of lifelong learning.

The Becta/DfEE software initiative has a number of objectives which include providing advice and guidance to teachers and the developer/provider community, with the aim of encouraging an increase in the availability of high quality software.

"We are carrying out research into the quality and range of the software currently available for schools, and disseminating this information to government, educators and the software industry."

The Curriculum Software Reports identify key issues in the provision of electronic resources for schools and enable planning for enhanced ICT use to support delivery of the National Curriculum. These reports were compiled following seminars which took place during October and November 1999. Contributors to the seminars included developers, teachers, LEA advisers, and representatives from Government departments, Government agencies and subject associations.

A report was provided for each of the following areas of the curriculum:

Art - English - Geography - History - Information and Communications Technology (ICT) - Literacy - Maths - Modern Foreign Languages - Music - Numeracy - PE - RE - Science

For each subject, the report provides information for teachers, managers and developers in the following areas:

- What does ICT offer to this subject?
- Minimum software requirements for delivering the National Curriculum
- A toolkit of software
- Generic software requirements and issues
- Content requirements and issues
- Data requirements and issues
- Other Issues
- Recommendations

In the following excerpts of the reports for Geography, Science and ICT 'modelling' emerges as a user requirement.

Curriculum Software Initiative Report: Geography (BECTa, 2000a)

General software requirements

Modelling and simulation proved to be an issue of some concern across the age ranges. There is a need for tightly focused simulations in a wide range of topics across the breadth of the subject in which the key variables can be controlled or manipulated so as to explore processes and model a range of outcomes. These would be open to use both as demonstrations in direct teaching, and as a context for individual/small group investigations.

There was discussion about some of the older software that had been produced in BASIC but which was no longer available which with modern visual representation and interfaces could be very good resources.

There was discussion about the availability at Key Stages 3 (ages 11-14) and 4 (ages 14-16) of modelling software in which teachers and pupils could explore, adapt and build their own models and simulations. This type of software should come with sample models appropriate for the age range. One group discussed the possibility of addressing qualitative issues using expert systems which could provide valuable support for investigations and geographical enquiry. It was felt by some that route planning software also had a place at Key Stage 2 (ages 7-11) and above.

Recommendation

Modelling/simulations were high on members' requirements, and there is a need for further research into the level of detail and the priorities for development.

<u>Curriculum Software Initiative Report: Science</u> (BECTa, 2000b)

General software requirements

Models and Modelling

Modelling proved to be an issue of some concern across Key Stages 3 (ages 11-14) and 4 (ages 14-16). There is a need for modelling packages in a wide range of topics across the breadth of the subject, in which the key variables can be controlled and/or changed to demonstrate and investigate scientific models. These would be open to use both for demonstrations in whole-class teaching, and as a context for individual/small group investigations. Primary colleagues pleaded for 'anything for Key Stage 1' (ages 5-7) which was appropriate.

Content requirements

Modelling

Teachers explicitly mentioned the need for modelling software, pointing to the value of 'doing modelling'. A popular area for a modelling program is one which demonstrates molecular models for chemistry at Key Stage 4 (ages 14-16) and beyond; these show how structure dictates a molecule's behaviour.

There are many other examples of science models in the Key Stage 3 (ages 11-14) scheme of work. Other suggestions include:

- pond life simulations
- populations
- photosynthesis
- balanced forces
- diffusion
- what atoms look like
- displacement reactions.

Internet/e-mail

Considering the Internet as a tool, a number of uses were identified. These include

- A site with lesson materials, worksheets and questions might be useful to some teachers.
- Conferencing, where a group of teachers sharing a common interest might interact.
- Uses of e-mail by teachers, possibly to extend good practice. A model was described where a teacher in one school tutored perhaps six other teachers in nearby schools, and they in turn trained more teachers.
- Use of e-mail by pupils, for example in sharing their data and other work with pupils in other parts of the country or the world.

Recommendation

- There may be a shortage of simulation and modelling software. There are some packages which are well received, but there are gaps and there is a need for a generic modelling package.

<u>Curriculum Software Initiative Report: ICT</u> (BECTa, 2000c)

Software requirements

Modelling software

There was considerable discussion on this area. It was felt that there was a need for a range of modelling software. The following could be used:

- spreadsheets
- simulations
- specific modelling software
- microworlds.

These needed to be able to cover quantitative and qualitative data and, if possible, semi-quantitative approaches. There were a number of issues, including:

- the need for a visual representation of model structures
- good feedback/response for users when building and testing models
- effective graphing and representation of output from models
- a range of pre-built models in all modelling software.

Spreadsheets are not very effective tools for modelling and used only in the absence of more suitable products. There is a need for some specific modelling software which will exemplify queuing theory.

Some of the old simulations and other software originally developed for the BBC and old RM machines should not be ignored, as some were based on effective learning models. The underlying principles could be used to update and improve modelling software.

Other software

Decision making and problem solving are fundamental skills for many areas of ICT and must be about promoting thinking skills. Modelling and simulation software are crucial for this area, as are good databases, and there is a need for software which helps students to structure their approach to problem solving.

3.3 Summary

To summarise the points that derive from the above surveys and reports and are informative for the design of the MODELLINGSPACE learning environment:

Concerning the use of ICT in education in the European context:

- Education policies in European countries are increasingly geared to the use of ICT.
- Overall, secondary schools are better equipped with computer facilities than primary schools.
- In many EU countries, even at primary level, learning about ICT has now become (or in some countries is becoming) an integral part of the compulsory school curriculum; at secondary level, this situation is virtually the norm.
- The objectives pursued by the teaching or use of ICT cover a broad range of skills, including the use of various software packages (one of which could concern modelling) and communication via a network.

The latter, and more generally the use of the Internet for educational purposes, however is still not very widespread in the EU schools. The main factors contributing to this situation include: poor levels of equipment (on average 25 pupils per on-line computer); slow connectivity (dominated by narrowband technologies); and lack of familiarity of pupils and teachers with the Internet (this reason was cited by between a fifth and a third of the teachers in France, Greece and Portugal).

- The use of ICT as a tool to be used for projects or for educational content is the approach most commonly recommended for primary schools. At secondary level, in addition to these uses, ICT is taught as a separate subject in almost all European Union countries.
- A majority of teachers at primary and general secondary levels claim to use ICT with their pupils in the course of their teaching on a regular basis. However, the numbers vary significantly amongst different countries in the European Union.

Concerning general requirements for modelling software (identified by teachers in the UK):

- Teachers of geography and science (in the UK) identified the need for modelling packages in a wide range of topics across the breadth of both subjects, in which the key variables can be controlled and/or changed so as to explore processes and model a range of outcomes. Science teachers in particular referred to the need for a generic modelling package.
- Modelling packages on the whole should be open both for demonstrations in whole-class teaching, and a context of individual/small group investigations. Moreover, the software should come with sample (pre-built) models appropriate for a given age range.

Some of the specific requirements teachers (in the UK) identified for modelling software are:

- to have modern visual representations of model structures;
- to have modern interfaces;
- to be usable by pupils and teachers to explore, adapt and build their own models and simulations;
- to model qualitative issues using expert systems;
- to cover quantitative and qualitative data, and if possible semi-quantitative approaches as well;
- to be usable by primary school pupils;
- to have good feedback/response for users when building and testing models;
- to have effective graphing and representation of output from models;
- to promote decision-making and problem solving skills.

Considering the Internet as a tool a number of issues were identified by teachers. These include:

- A site with lesson materials, worksheets and questions for teachers to use.
- Electronic conferencing amongst teachers.
- Uses of e-mail by teachers, to extend good practice.
- Use of e-mail by pupils, to share data and work with pupils in other parts of the country or the world.

4. Review of Technology Based Modelling Environments

4.1 Categorization of modelling environments according to the corresponding formalizations and conceptual framework

We can categorise the modelling software environments into three categories depending on the kind of reasoning and modelling they promote:

1. **Quantitative modelling software**

   Quantitative models make use of quantifiable variables and algebraic relations. One can create or explore such models. The big majority of existing modelling software belongs to this category.

2. **Qualitative modelling software**

   Qualitative models express relationships which cannot be expressed in a quantifiable way, and of which the criteria of validity are not strictly defined. Such relationships appear in abundance in the school curricula. Almost all non-quantitative modelling software belong to this category (e.g. some expert systems, concept maps, etc.).

3. **Semi-quantitative modelling software**

   Semi-quantitative models involve quantifiable variables, whose change however is not defined by algebraic relationships, but by the kind of influence that one exerts on the other. In other words these models are concerned mainly with qualitative relationships. There are only few of these software available in the market; most of the existing ones have been developed and are used for research purposes.

The uses that a modelling software can be put in can be further grouped under two axes (see Table 4.1):

a. expression (involving creation of new models)

b. exploration (involving exploration of ready-made models, through their simulation)

Kind of modelling	Expression	Exploration
Quantitative	Systems of mathematic models, having as basic tools mathematic equations and spreadsheets	Exploration of scientific simulations, equations.
semi-quantitative	Construction of qualitative models of relationships between factors (dependent and independent variables).	Qualitative simulations of results and relationships between factors and variables, based on order of magnitude and on relation between possible values.
Qualitative	Expert systems, concept maps, semantic networks, story generators, adventure games.	Decision making and taking simulations, logic relationships.

Table 4.1: General categories of kinds of modelling software and their uses.

We will now focus on modelling software which has been designed/developed or used for educational purposes. Table 4.2 presents the 'profile' of a number of such software, i.e. the year of its release (symbol '->' signifies that the product is continuously updated), its provenance (names of authors/developers, and kind of organisation: company or research institute/university, or both when it was originally created as a research tool and then became a commercial product), its status (prototype or commercial) and finally its target audience.

Modelling Software	Release Year	Provenance	Product status	Target Group
TOOLS FOR QUANTITATIVE REASONING				
Spreadsheets (e.g. EXCEL, LOTUS, etc.)	*Different editions*	*Different companies*	Commercial product	Students in lower-and upper secondary schools. Adults
Q-MOD	1990	Institute of Education, King's College, Tools for Exploratory Learning Programme	Research tool	Pupils 11-14 years old
MODEL-BUILDER	1991	The Advisory Unit for Microelectronic for Education, Kings College, London/ Mary Webb	Research tool	Pupils 14-15 years old
PROBSIM (Probability Simulator)	1992	Scientific Reasoning Research Institute, Univ. of Massachusetts / Peter Wilder	Commercial product	University students
DMS [Dynamic Modelling System]	1985, 1991	Chelsea College (King's College) / J. Ogborn. Published by Harlow Logman Micro-software	Commercial product	Pupils in upper secondary schools and university students.
DYNAMO [Programming Language)	1983	Roberts, et al, 1983	Research tool	University students
CMS (Cellular Modelling System)	1987	Institute of Education (Univ. of London) / J. Ogborn.	Research tool	Pupils in upper secondary schools and university students
STELLA	1985-1997 →	HPS / A. Pytte, J. Doyle	Commercial product	Scientists, university students, upper secondary pupils.
EXTEND	1987	Imagine That, Inc., California / B. Diamond	Commercial product	Pupils in lower and upper secondary schools.
ALGEBRAIC PROPOSER	1987	True Basic, Inc, Hanover / J. Schwartz	Commercial product	Pupils in lower and upper secondary schools.
NUMERATOR	1989	Cambridge, Logman Logotron / P. Hunter	Commercial product	Pupils in upper secondary schools

MODELLUS	1997, →	Knowledge Revolution / D. Teodoro	(1995-2002) Research tool (1997-1999) Commercial product	Pupils in upper secondary schools, university students.
SIMQUEST	1998, →	SERVICE consortium University of Twente, The Netherlands	Research tool	Vocational training pupils, Students, Adults
TOOLS FOR SEMI-QUANTITATIVE REASONING				
IQON	1990, 1993	Institute of Education, Tools for Exploratory Learning Programme	Research tool	Students in lower-and upper secondary schools, University students.
LinkIt (WlinkIt for Windows)	1996	Institute of Education/ F. Sampaio	Research tool	Students in lower-and upper secondary schools
Model-It	1995, →	University of Michigan GoKnow, LLC	Research tool Commercial product	Students in lower-and upper secondary schools
TOOLS FOR QUALITATIVE REASONING				
A. Semantic networks and concept maps				
AXON Idea Processor	1994 - 2001 → (Axon2002 R2.52 released)	AXON Research, Chan Bok	Commercial product	Scientists, Planners, Writers, Managers, Pupils, Adults
Activity Map	1994 1997	Time/System Int.	Commercial product	University students, Adults
Belvedere	1994 1998 (v. 3.0)	University of Pittsburgh	Research tool	University students, school pupils
CLASS	1998	Heriot-Watt University (R. McAleese)	Research tool	Students
Cmap	1997 2001 (2nd ed.) →	IHMC (J. D. Novak)	Research tool Freeware	Scientists, Planners, Writers, Managers, Pupils, Adults
Decision Explorer	1991 2001 (v. 3.1) →	Banxia Software	Commercial product	Single user or group work (commercial or academic) Secondary education pupils, Adults
Inspiration / Education edition	1991 2001 (6th vers.) →	Inspiration Software, Inc., USA	Commercial product	School pupils, University students, Adults
Kidspiration	2001, →	Inspiration Software, Inc., USA	Commercial product	Pupils, students K-5
Kmap for mac	Not available	Mildred L. G. Shaw Brian R. Gaines University of Calgary, Alberta	Research tool	Scientists, Professionals, Higher Education
Mind Manager	1994 2002, →	By Michael Jetter	Commercial product	University, Adults

Mind Mapper	1997 2002 (v.3.2), →	Sim Tech Systems	Commercial product	Secondary education, University, Adults
Mind Mapper Junior	2002 (v.1.5), →	Sim Tech Systems	Commercial product	Pupils, students K-5
MOT	2001, →	Université de Montréal	Freeware	Pupils, Teachers, Students
PIViT	1993 1996	PBS group at the University of Michigan	Research tool Commercial product	Project Based Science pupils, Teachers, mathematics, English/language, arts
Representation Tool	2000	IACM/FORTH	Research tool	Project Based Science pupils, Teachers, Students
SEMNET	1983 1991	Sem Net Research Group, San Diego California	Commercial product	Secondary education pupils, University students, Adults
Smart Ideas	1996 2002 (beta version), →	SMART Technologies	Commercial product	Teachers (Distance education, technology classrooms), Trial participants, Journals
VisiMap (v.2.5)	1992 1998, →	Coco Systems	Commercial product	University students, Adults
B. Expert Systems				
ENERGY EXPERT	1991-1994	Kings College, London Modus Project	Commercial product	Students in primary and lower-secondary schools
EXPLORE YOUR OPTIONS	1989-1993	Institute of Education, London/ Tools for Exploratory Learning Programme	Research tool	Pupils 11-14 years old
KNOWLEDGE PAD	1989	Kingston Polytechnic, UK, J. Briggs.	Research tool	Students in lower-and upper secondary schools
LINX88	1988	Institute of Education, / J. Briggs	Research tool	Students in lower-and upper secondary schools
SUPERLINX	1990	Institute of Education, / J. Turner	Research tool	Students in lower-and upper secondary schools
PICTURE SIMULATOR (objects and rules)	1991	Newsoft Group, Portugal/ A. S. Camara,	Commercial product	Adults
C. Systems based on AgentSheets technology				
WORLDMAKER	1992-94 1995, →	Institute of Education, London/ Richard Boohan, Jon Ogborn and Simon Wright University of Hong-Kong/ Nancy Law	Research tool	Pupils 11-14 years old

Table4.2: Educational modelling software

Table 4.2 allows us to make some observations:

- A significant number of educational modelling software have originated and have remained as research tools. More particularly, 11/25 of the qualitative, 2/3 of the semi-quantitative and 5/13 of the quantitative modelling software are not commercial products, despite the fact that in recent years (last decade) there has been an increased interest in the commercial development of modelling systems.
- There are only three modelling systems which support semi-quantitative reasoning, and only one of them is commercially available.

The systems that are of particular interest to us are those which fulfill the following criteria:

- They target pupils of 13-15 years of age.
- They support all three kinds of reasoning, i.e. qualitative, quantitative and semi-quantitative reasoning.
- They can support modelling of situations in physics, mathematics, environmental education.

None of the above software fulfills all three criteria.

STELLA supports mainly quantitative modelling and has been designed mainly for adults. Model-It supports both quantitative and semi-quantitative reasoning, but can model a restricted category of situations, mainly of environmental interest.

IQON (or the more recent and developed version LinkIt) and Q-MOD, being the products of the same research programme, could be combined in order to cover semi-quantitative and quantitative modelling. However, they have only the status of prototype; do not make use of any multimedia content; support the creation of quite abstract models; and offer very limited options of alternative representations.

Having established that none of the available modelling software has all the functionalities that we aspire MODELLINGSPACE to have, we proceed with a more detailed analysis of some of them, thought to be of more interest, because they have some relevant features and/or are more commonly used.

4.2 Analysis of modelling environments allowing creation of quantitative models

As mentioned in section 1.10.4 and above all quantitative models use algebraic relationships to calculate the values of the dependent variables from those of the independent variables, which are initially specified. So, the behaviour of the model depends on both the values of the independent variables and on the nature of the relationships between the variables.

There is need to distinguish between two fundamentally different kinds of quantitative model: 'static' and 'dynamic' (Boohan, 1994a). In a static model, initial values are given to the variables, and output values are calculated. In a dynamic model, the calculated values are fed back into the relationships so that the behaviour of the system is modelled over time.

Some modelling environments can support static models, others are better for dynamic modelling and some support both. In the following sections we will review a selection of them which are more interesting for our purpose, that is which have features we may wish to incorporate (or avoid) in the MODELLINGSPACE environment.

But before going on to discuss some modelling tools which are either commercially available or are products of research programmes and have been used in education, it is worth saying few things about the case of spreadsheets.

4.2.1 Spreadsheets for quantitative modelling

Spreadsheets have been increasingly used in the school curriculum as a modelling tools. Being a general-purpose software, which is commonly available in schools, explains their popularity. They have been developed for commercial use, and this makes them robust with a level of sophistication which is difficult to match in products developed for the educational market. They are also very versatile and suitable for many kinds of models, e.g. static and dynamic; deterministic and probabilistic (Boohan, 1994a; Brosnan, 1994).

Essentially a spreadsheet consists of an array of cells which can have values. A value may be entered directly in a cell, thus representing an independent variable, or it may be calculated by entering a formula in the cell, thus representing a dependent variable. The ease with which one can enter new values renders them ideal to be used to model static quantitative problems, and to explore 'what-if' questions and trial-and-error solutions. Moreover, the fact that the model is explicit (pupils can see the equations) and can be changed by the user, can potentially be very beneficial to pupils in their understanding of concepts and relationships.

Although spreadsheets have many advantages, they were not designed as modelling tools and can be difficult for pupils to use. They are limited visually, and links between variables cannot be represented. Moreover, they do not allow for 'real-time' interactions with the model, by adjusting the sizes of variables while the model is still running. 'Real-time' interaction with a model is important, since this is the way we are used to interact in 'real-world' situations. Various purpose-built modelling software address these requirements.

4.2.2 Analysis of STELLA modelling tool

This section has two subsections. The first subsection gives a general description of STELLA (Structural Thinking Experimental Learning Laboratory with Animation) focusing on its building blocks and interface. This description is based on the writings of the following authors: Greenman, 1994; Mandinach, 1989; Richmond, 1987; Sampaio, 1996; Santos, 1992; Steed, 1992; Whitfield, 1988.

The second subsection discusses critically the software assuming its use in an educational environment.

The STELLA version considered here is 7.0 for PC[11].

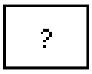
4.2.2.1 STELLA: Background

STELLA is a simulation package first introduced in 1985 and published by High Performance Systems of Hanover, New Hampshire [Richmond *et al.*, 1985, 1987]. A major objective in developing Stella was to provide the business community with the tools to create and run system dynamics models. This methodology was the outcome of work initiated in the 1960s and 1970s by Jay Forrester [see Forrester, 1968] and colleagues at the Massachusetts Institute of Technology - applying control theory concepts to the construction of models of socio-technological systems.

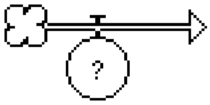
4.2.2.2 STELLA: General Description

STELLA is a computer system designed for the representation of dynamic models using iconic levels, flows and converters. In a certain way it can be seen as an implementation of DYNAMO (an acronym for DYNAmic MOdelling) [see Roberts *et al.*, 1983] with a graphical interface. The system takes the idea of causal loops and elaborates on them through a metaphor of tanks, pipes and flows, providing the user with five building blocks:

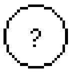
[11] STELLA was until 1994 only available on Apple Mac computers but Windows versions have since been released. STELLA 7.0 for PC platform is now available as a free Demo version. It can be downloaded from the web site: http://www.cognitus.co.uk/academic-solutions.html.



Stock - Represents a quantity that can increase (accumulate) or decrease (de-accumulate) over time. Stocks can accumulate both physical and non-physical staff.

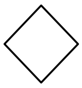


Flow - Controls the rate of incoming and outgoing material from stocks. A Flow can come from and/or go to a 'cloud' which means that its source or destination are not specified (it can mean that the cause or effect of the flow is not relevant to the model). Flows can be altered by stocks and converters.



Converter - Converts inputs into outputs. They can be constants or calculated from other quantities.

Connector - Connectors are used to determine that one variable in the model depends on one or more other variables in the same model. It is possible to distinguish between information and action linkages among model entities.



Decision Process Diamond (DPD) - DPDs enable you to encapsulate decision logic in a single object.

STELLA has three layers/levels (with distinct windows) at which one can work in parallel: the map/model level which contains the diagrammatic form of the model (see Figure 4.1) and on which the application opens by default; the interface level which contains the key components of the model and their interconnections but none of the detail; and the equations level, which contains the complete set of equations corresponding to the diagram. At the interface level the user feeds in information through a range of hypertext-like devices (including sliders, fields and buttons) and the output can be superimposed on the relevant map components. Movement between the different layers is achieved with a single mouse click if the *author* of the model so permits.

One attractive feature of the interface layer is the ability to import still pictures and movies. The significance of these hypertext and multimedia facilities is that "flight simulator" environments can be constructed in which the user can imagine him or herself influencing the system through various control devices.

Figure 4.1 below shows a model about population dynamics constructed with STELLA. There is also a bottom window showing the (partially complete) mathematical equations that describe the model.

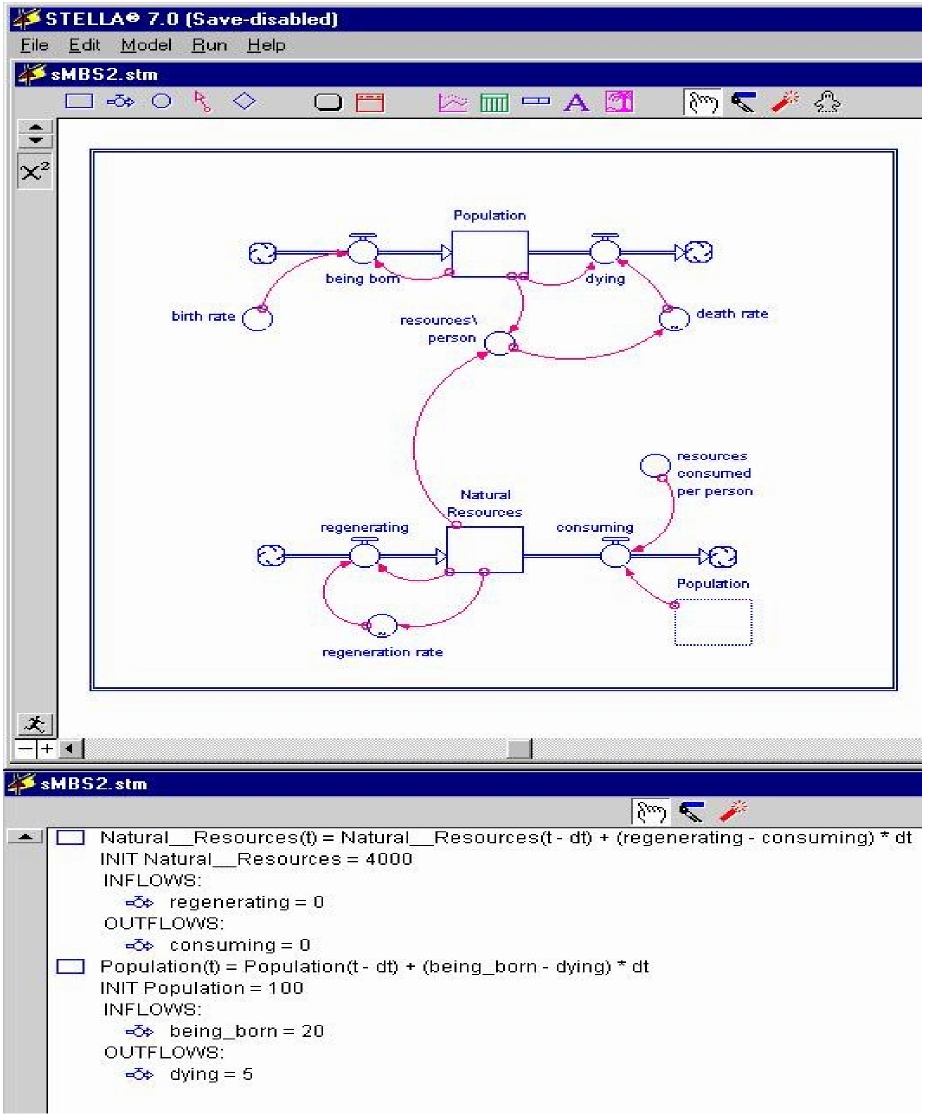


Figure 4.1: STELLA: Model about population dynamics with its set of equations

The control panel on the top of the window provides the elements to edit a model. The first five icons represent the five different building blocks provided in STELLA. To create a stock or converter, the user has to select one of the blocks (single click) and point and click on the place he/she wants to place the object on the working area. The middle seven icons are called 'objects' and are: the 'button' to which one can assign various roles, such as of executing menu commands, or of providing instructions for how to run a simulation, of navigation, playing movies, revealing model structure etc.; the 'sector frame' which is useful for partitioning of parts/sectors of the model for visual clarity, and for simulating parts in isolation from the remainder of the model; the 'graphs pad' and 'table pad' for presenting the results in graphical or tabular form, the 'numeric display'; the 'text box' and the 'graphics frame'.

Finally, the last four icons correspond to the tools to manipulate the building blocks: the first icon (the hand) is used to select and move items; the second icon is a paint brush which can be used for colouring objects, frames or text; the third icon (the dynamite) is used to erase the building blocks; and the fourth icon (the ghost) duplicates an element that can be placed anywhere on the window; the last icon, the dynamite, serves to break existing connections and variables.

After drawing the model the user has to double click on the objects on the screen to set the equations and the initial conditions of the variables. After that it is necessary to set the scales (range of variation) for the objects created and run the model (both functions are inside the *Run* pull-down menu). Another way of relating variables of the system is through a table or a graph (drawn by the user) that specifies the relation between two variables of the model.

The outputs of STELLA can be of four kinds:

- Animated diagram - The graphics capabilities of the system permit the user to see in real time the flow of information in the model. The levels of the stocks move up and down representing a tank being 'filled' or 'emptied'. Flows and converters have small arrows that move across their icons as a function of their value.

- Table pad - Show in a table the values of the objects chosen by the user.

- Graph pad - Show a graph of the objects chosen by the user. The system can show scatter plots of two variables or a time series graph of up to four different variables.

- Numeric display of results.

One crucial part of simulation with Stella is sensitivity analysis. Once one has constructed a model and fitted it with numerical values, one can test the sensitivity of model outputs, and the conclusions drawn from those outputs, to variations in the model's parameters.

According to the introduction to the software, which is provided together with the Demo version 7.0 of STELLA, the STELLA program isn't just useful for constructing and testing models, but is also useful for sharing the logic of a model with others. It makes it easy to provide feedback to people interacting with underlying models, with the aid of the "Flight simulator" interface.

"In a Flight Simulator, feedback comes in the form of messages *that alert people to conditions in the system with which they are interacting, and* coaching sequences *that help people to understand how they are creating the results that they are creating."*

Another key 'sharing' feature of the STELLA program is called 'storytelling'. It enables one to 'program' the unfurling of the structure/behaviour of a model, so that people not involved in constructing the model can build an understanding of its logic, and associated dynamic behaviour, one piece at a time. Also, one may choose to annotate the unfurling of a story by associating text, sound, movies or graphics with the appearance of a model variable, or group of variables.

4.2.2.3 Mathematics of STELLA System

After creating a diagrammatic representation of a problem on the screen, the user has to describe how the elements presented on the diagram change in time and in respect to the values of other variables and constants.

In order to do so the user has to go to each element of the model and mathematically express them in terms of their causal factors using finite algebraic relations (relations for 'stocks' are interpreted by the system as finite difference equations). At this stage the system gives some help to the user by presenting a set of elements (other variables/constants existent in the model) that are graphically connected to the variable in question. Figure 4.2 presents a window requiring the mathematical definition of the variable *being born* (shown in the model in Figure 4.1). Its only causal factors – *Population* and *birth-rate* - appear in a small window called "Required Inputs" and can be used to describe the dynamic behaviour of *being born.* The system also provides a set of mathematical functions that can be used by the user. They appear in the window called "Builtins".

After specifying the mathematical relations among the elements of the model the system can simulate it by solving the finite difference equations using one of three possible numerical methods (which can also be chosen by the user): Euler, 2nd-order Runge-Kutta and 4th-order Runge-Kutta.

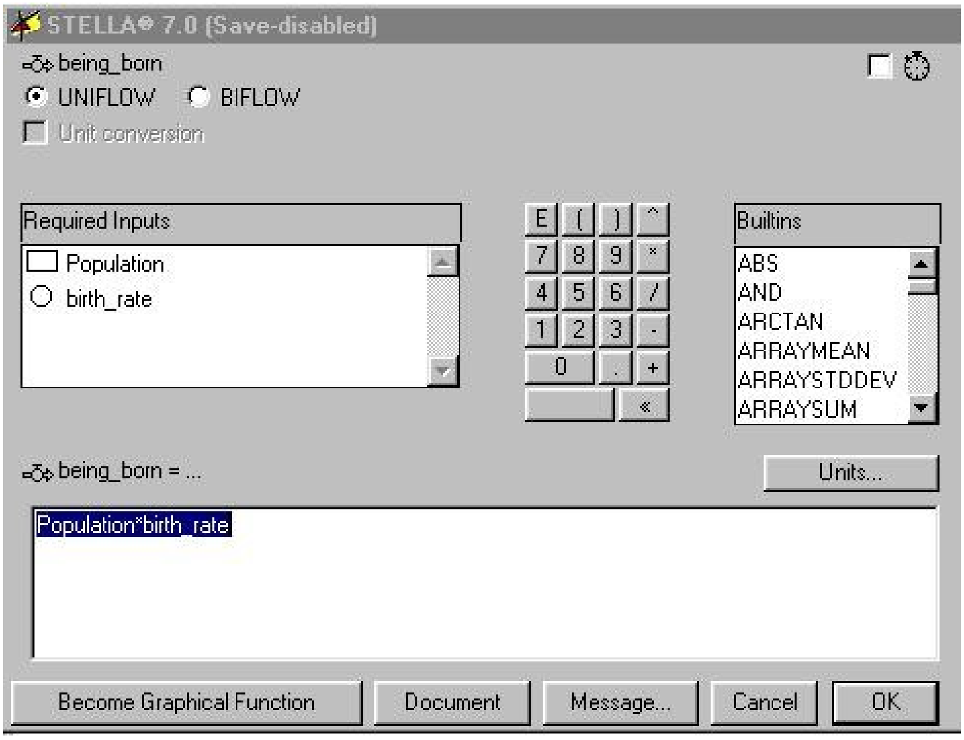


Figure 4.2: Defining the mathematical computation of a variable in STELLA

The scope of Stella modelling is further extended through the use of additional icons representing discrete events and the inclusion of logic and stochastic functions in the extensive library of in-built functions.

4.2.2.4 Pedagogical theses

In the Guide 'STELLA introduction to Systems thinking' [Richmond *et al.*, 1997] which accompanies the STELLA software, STELLA's authors suggest that there are five learning processes which encompass much of what goes on learning-wise both within and outside the formal education system. These five are:

- Assimilating Content;
- Gaining Understanding;
- Building Understanding
- Building the Capacity for Building Understanding; and
- Building the Capacity for Sharing Understanding.

In relation to these five learning processes STELLA's authors claim that *"It is the explicit aim of both Systems Thinking and the **STELLA** software to support the three 'building' activities."*

In the following excerpt they explain how STELLA aims to achieve this aim:

*"First, Systems Thinking applied in conjunction with the **STELLA** software - infuses discipline into the process rendering of mental models. The discipline helps to squeeze out ambiguities and internal inconsistencies which otherwise might cloud the mental model.*

Second, by enabling mental models to be made explicit in a visually precise language, other people can examine them. The resulting collaborative learning process - benefiting from a diversity of perspectives - surfaces and fills in the blind spots which inevitably populate mental models whose assumptions remain implicit.

Third, the STELLA software enables students to rigorously test their mental models to determine whether they can in fact be accorded the status of "entertainable hypotheses". In so doing, the software is closing a vital loop in the process of building understanding - a loop which in many circumstances would otherwise remain open.

Fourth, through continued use of the STELLA software to render and test mental models, the general capacity for both building understanding and performing mental simulation is augmented.

And fifth, using the software to develop Learning Environments hones students' empathic capabilities the capacity for sharing understanding with others."

4.2.2.5 Curriculum use of STELLA

STELLA is considered as an across-the-curriculum modelling tool.

The range of applications possible in STELLA is illustrated by the various sample models that have been bundled with the demo version of the software. They address a variety of disciplines:

- *Biology and Environmental Science:* Pollution dynamics; Predator-prey dynamics
- *Science and Research:* Nitrogen cycle; Pharmacokinetics; P-I controller; The Balloon problem, etc.
- *Physical Science and Engineering:* Chemistry: Le Chatelier's principle and reversible reactions; Physics: Hooke's law and Virtual Bungee jumping
- *Social Science:* Tobacco use prevention policy design laboratory; Immigration dynamics
- *Literature and Philosophy:* Virtual Hamlet
- *Mathematics:* Distance and time problems; Related rates.

Greenman (1994) also reports that STELLA featured prominently in a Department of Trade and Industry sponsored project for secondary schools in the UK called *"Modelling across the Curriculum*":

"In this project Stella simulated a variety of situations in the natural sciences including oscillatory behaviour in mechanics, enzyme reactions in chemistry and soil drainage in environmental science. Spreadsheets were also used in the project, modelling dynamical behaviour in both the social and natural sciences, in order to illustrate and contrast the spreadsheet way of doing things. It is clear however that the flexibility, ease of use and the complexity that is achievable with Stella gives Stella a definite advantage in simulation."

4.2.2.6 Discussion of STELLA System

The following discussion is partly based on relevant discussion written by Fabio Sampaio [1996] in his PhD thesis.

A beneficial aspect of STELLA is its exploitation of a graphic interface to represent and animate the structural diagram of a given problem. With such a system it is possible for the learner to visualise changes to the whole system over time, permitting him/her to have a better understanding of phenomena.

Another positive aspect of STELLA is the different outputs it can present for a certain simulation. Discussing and linking the ideas embedded in these different outputs can help the process of creating meaning from different representations [Teodoro, 1994].

However, in respect to the interface, three main problems can be pointed out. First, the objects that represent the variables of the problem being modelled (stocks, flows and converters) do not present the initial quantities attached to them by the user until he/she asks for the system to run the model. This can be cumbersome for the learner because he/she cannot have a qualitative idea about the initial state of the model before starting a simulation.

The second problem is that although the system permits the user to set individual scales for the variables, it does not explicitly show these scales when the software is running. Again this can confuse the user especially in situations where he/she has many objects on the screen.

Finally, another confusing feature of the more recent version of STELLA is that a flow can be "uniflow' or "biflow". A "uniflow" is a flow that it can have only positive values (things entering the stock) and a "biflow" is one that can have both positive and negative values (things entering or leaving the stock). By default, flows are "uniflows" – this means that in almost all physics models one must be very careful and not forget to change from "uniflow" to "biflow".

The next two problems are more fundamental and have to do with how the system was conceptualised and with how the visual metaphor is used to represent models.

An inconsistency between the mathematics and the visual metaphor of the system is that although it is possible to have negative values in the underlying model (at the mathematical level), the minimum visual value for a stock on the screen is zero ('empty'). If we recall that STELLA was developed for students being initiated into modelling, then the visual representation of a model must have much more importance than the mathematics that goes behind it.

It is also true that the visual metaphor used by STELLA is not equally appropriate for all kinds of problems. For instance *velocity* and *displacement* are two physical entities not at all obviously appropriately represented by a tank, though in STELLA they need to be. Although they can increase and decrease, people do not normally associate them with the idea of accumulating or de-accumulating. In the same way, the systems that can most easily be modelled are those directly associated with its visual metaphor, such as hydraulic systems or economic systems (with stocks and flows).

Another negative aspect of STELLA is the fact that it is out of the reach of younger students due to the mathematics required to create models. Although phenomena involving simple algebraic functions and first order equations can be picked up by 12-15 years old students, it becomes difficult to use the system if the mathematical model has to involve higher order equations and/or quadratic, logarithmic and sinusoidal (etc.) functions. In this case the only possibility open to the students is to interact with simulations already created by their teachers.

4.2.3 Analysis of MODELLUS modelling tool

The MODELLUS version considered here is version 2.01.

4.2.3.1 MODELLUS: Background

MODELLUS is a modelling package first introduced in 1997 and currently published by Knowledge Revolution (Teodoro, Vieira and Clérigo 1997; 2000).

4.2.3.2 MODELLUS: General principles of design

'Modellus' concept can be briefly described as:

1. It is a software tool to create and explore multiple representations of mathematical models using functions, differential equations and iterative equations.
2. It has a multiple window environment. In one of the windows, the user can write a model, writing equations as they are written on paper; in other windows, the user can create and interact with animations of the models, using abstract objects, such as vectors and graphs, or more concrete objects, such as video and photos.
3. The communication with the user is based on the concept of "intellectual mirror" (Schwartz, 1989)—the software acts as a mirror of what the user think.

Same as all the software now available, 'Modellus' has a direct manipulation interface. However, A fundamental design choice made in 'Modellus' was to identify what could be done using graphical metaphors and what should be done using other representations. Some designers choose graphical metaphors, even for equations (e.g., STELLA, at least partially; Feurzeig, 1993). In 'Modellus', the choice was different: the user should write equations as he writes on paper (as far as possible). The environment for manipulating equations and all other features are implemented in standard graphical Windows elements: windows, pull-down menus, dialogue boxes, buttons, etc.

Another important design choice was that the user should be able to interact with animations of the models when they are running. This interaction can be done in all variables that are independent (such as parameters) or integrated. The user cannot interact with dependent variables defined as functions. This feature, as well as others, will be explained below in more detail.

4.2.3.3 The structure of the program

After running 'Modellus', the screen looks like the following (Figure 4.3):

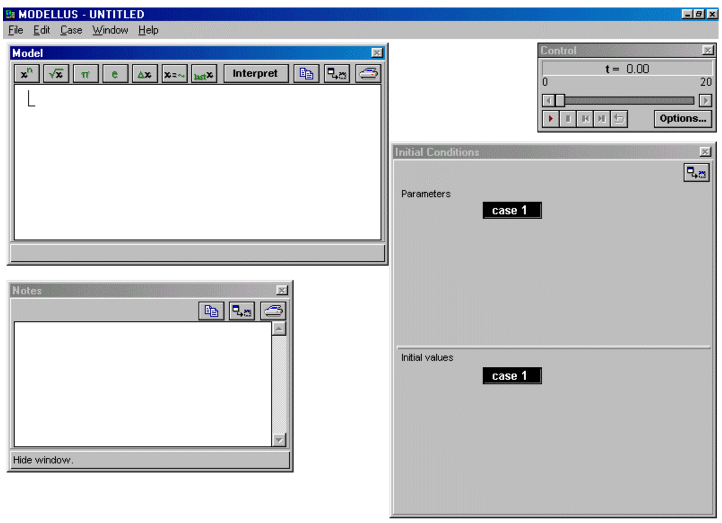


Figure 4.3: Modellus appearance after run

'Modellus' has a main window and the following parent windows:

1. a *Model Window*, to write and edit mathematical models or comments;

2. a *Control Window*, to start, pause, and stop the model and also to control general specifications of the model, such as the letter and the domain of the independent variable;

3. a *Notes Window*, to write notes and comments;

4. one to three *Graph Windows*, available through the Window menu;

5. one to three *Table Windows*, also available through the Window menu;

6. and one to three *Animation Windows*, available through the same Window menu.

4.2.3.4 Creating and running a file

To create and execute a 'Modellus' file the user must write one or more functions (or differential equations, or difference equations) on the *Model Window*, create an output window (a graph, a table or an animation) and run the model using the start button on the control window.

For example, to create a model of an oscillator, using a function, a system of differential equations, or a system if iterative equations, the user can write one of the following models (Figure 4.4) on the *Model Window*.

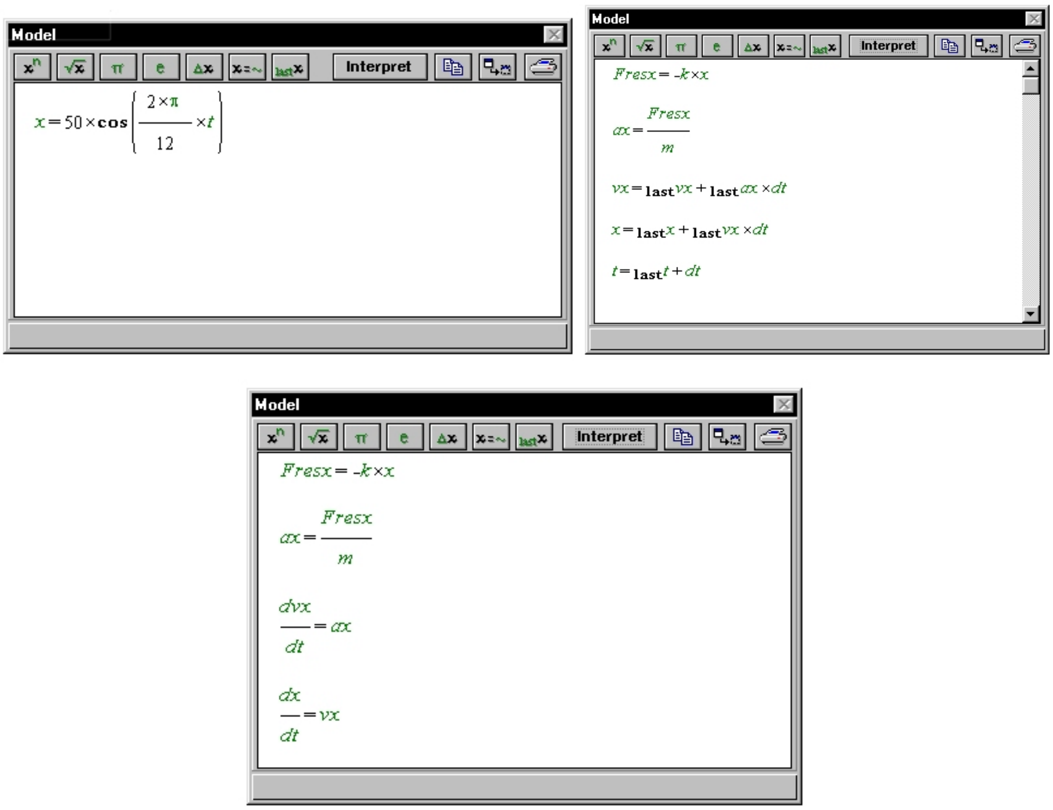


Figure 4.4: Three ways of writing a model of an oscillator on the *Model Window*

Equations look like a normal written equation. The parser formats equations using a few conventions: variables are written in *green italic*, primitive functions in **black bold**, numbers and operation symbols in plain black. To get the symbols of the algebraic operations the user must press the usual keys (e.g., the * for the multiplication--this operation can also be obtained with the space bar). Units are not represented on the model but the user must always be aware of them if that is the case. Angles can be expressed in degrees and radians. By default, 'Modellus' uses degrees, but this can be changed pressing the *Options...* button on the *Control Window*.

Once the model is written, it is necessary to press the *Interpret* button to activate the parser and check if the syntax is correct. This action is also automatically performed when the user press the start button on the *Control Window*.

To see an output of the model the user must activate another window, such as a *Graph*, a *Table* or an *Animation*. *Graph* and *Table Windows* are very easy to create through the *Window* menu. It is possible to see one or more variables on the *Graph* or *Table* windows, clicking on the names of the variables with the Ctrl key down (a standard procedure for multiple selection on Windows) (Figure 4.5):

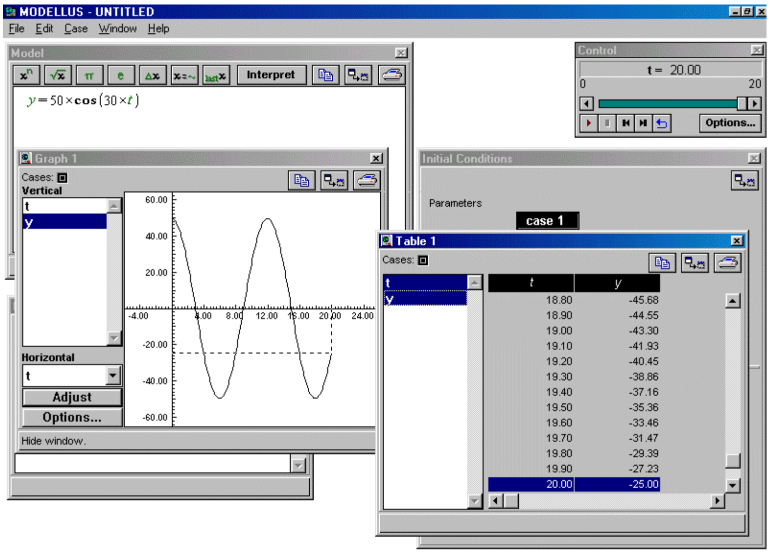


Figure 4.5: Two output windows: *Graph* and *Table*

It is possible to resize any window, which is particularly useful for the *Graph Window*. It is also possible to *Adjust* automatically the scale on the *Graph Window* (button *Adjust*) and choose different viewing options. For example, the user can choose *Equal scales*, *Projection lines*, *Points* instead of line, etc. pressing the button *Options...* on the *Graph Window*. Other actions on this window include zoom in (click and drag) and change the origin (double click).

After making visible an *Animation Window*, using the *Window* menu, it is possible to create a visual representation of the model--in this example, an oscillating object. Any *Animation Window* has a set of objects that can have properties, such as position or size, accordingly with the available variables. For example, to create an oscillator, the user can choose a particle, clicking on the corresponding button, and attribute the values of variable *y* to its vertical coordinate (Figure 4.6).

Once created the particle, its motion is animated when the model is running. The graph is generated simultaneously with the motion.

The *Animation Window* can have many types of objects. For example, in Figure 4.7 the *Animation Window* has a particle, a vector, a graph, and a block of text and digital displays of position and time.

Objects (position of the particles, length of the vectors, limits of the analog and protractors displays, graphs, etc.) on the *Animation Window* behave in accordance with a scale. By default, the scale is 1 pixel to 1 unit- in the above animation, the scale of time t, on the graph, is 1 pixel to 0.1 unit. To make a useful animation it is necessary, frequently, to change the scale of one or more objects.

It is possible to write symbols to represent parameters on the model. A set of values of the parameters is a Case. Using the Case Menu, the user can create a new Case. This is particularly useful to analyse the effect of changing the values of the parameters.

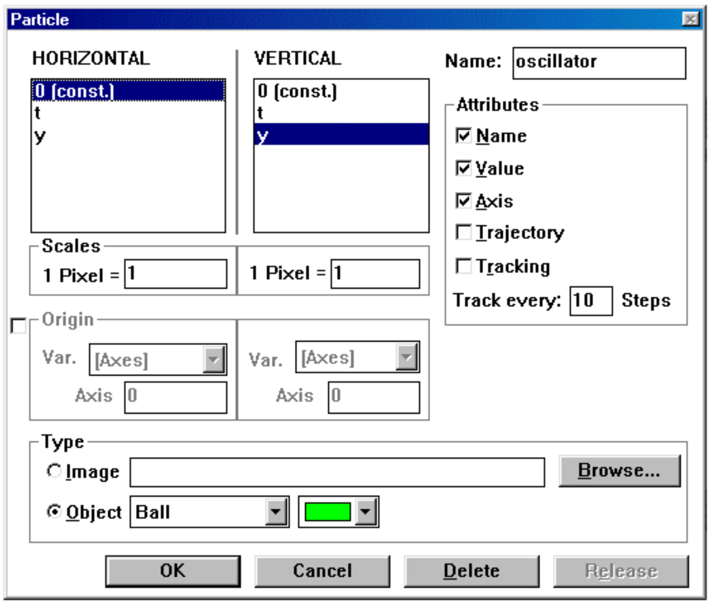


Figure 4.6: Properties of the oscillating particle: the vertical coordinate is y and the horizontal coordinate is 0.

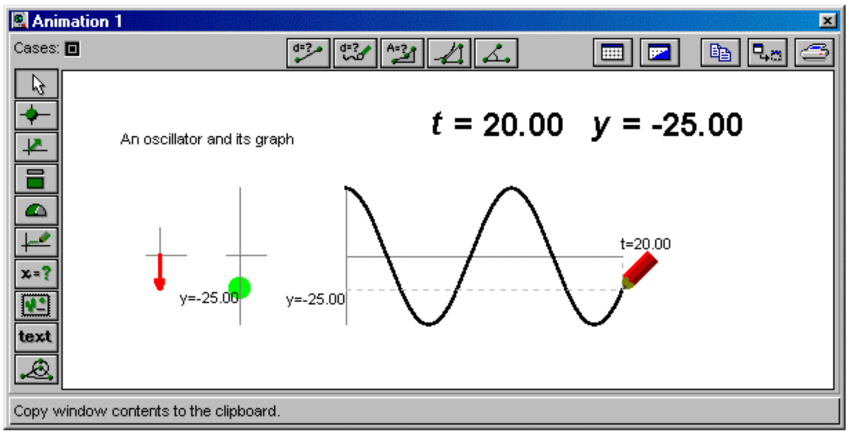


Figure 4.7: An Animation Window with different types of objects: text, vector, particle, graph, digital displays.

4.2.3.5 Mathematics of 'Modellus'

'Modellus' has a "motor" to compute symbolic derivatives of functions that explicitly depend on the independent variable or are derivatives themselves.

As usual, if the function is a trigonometric function, its argument must be in radians.

'Modellus' can use any step and any single letter for the independent variable. To change the default letter t and the default step 0.1 (and, if necessary, the lower and upper limits of the independent variable) it is necessary to press the *Options...* button on the *Control Window*. For example, it is possible to analyse the succession with limit e = 2.718... with a suitable model.

'Modellus' can also solve ordinary differential equations (ODEs), or systems of ordinary differential equations. A system of n differential equations,

$$\frac{dy_1}{dt} = f_1(t, y_1, y_2, \ldots, y_n)$$
$$\frac{dy_2}{dt} = f_2(t, y_1, y_2, \ldots, y_n)$$
$$\ldots$$
$$\frac{dy_n}{dt} = f_n(t, y_1, y_2, \ldots, y_n)$$

is solved iteratively, using Runge-Kutta fourth-order method with fixed step dt,

$$y_{n,t+dt} = y_{n,t} + \frac{1}{6}dt(k_{1,y_n} + 2k_{2,y_n} + 2k_{3,y_n} + k_{4,y_n})$$

where

$$k_{1,y_n} = f_1(t_t, y_{1,t}, y_{2,t}, \ldots, y_{n,t})$$

$$k_{2,y_n} = f_1\left(t_t + \frac{1}{2}dt, y_{1,t} + \frac{1}{2}dt \cdot k_{1,y_1}, y_2 + \frac{1}{2}dt \cdot k_{1,y_2}, \ldots, y_{n,t} + \frac{1}{2}dt \cdot k_{1,y_n}\right)$$

$$k_{3,y_n} = f_1\left(t_t + \frac{1}{2}dt, y_{1,t} + \frac{1}{2}dt \cdot k_{2,y_1}, y_{2,t} + \frac{1}{2}dt \cdot k_{2,y_2}, \ldots, y_{n,t} + \frac{1}{2}dt \cdot k_{2,y_n}\right)$$

$$k_{4,y_n} = f_1(t_t + dt, y_{1,t} + dt \cdot k_{3,y_1}, y_{2,t} + dt \cdot k_{3,y_2}, \ldots, y_{n,t} + dt \cdot k_{3,y_n})$$

With this method it is possible to solve the most common ODEs or systems of ODEs. For models with rapid changes in dependent variables, it can be necessary to choose smaller steps.

4.2.3.6 Modelling options in 'Modellus'

An important feature of 'Modellus' is that all integrated variables can be changed when the model is running. Consider, for example, a particle with mass m where a constant horizontal force is applied (component F_x). A 'Modellus' model for this phenomenon can be the one in Figure 4.8.

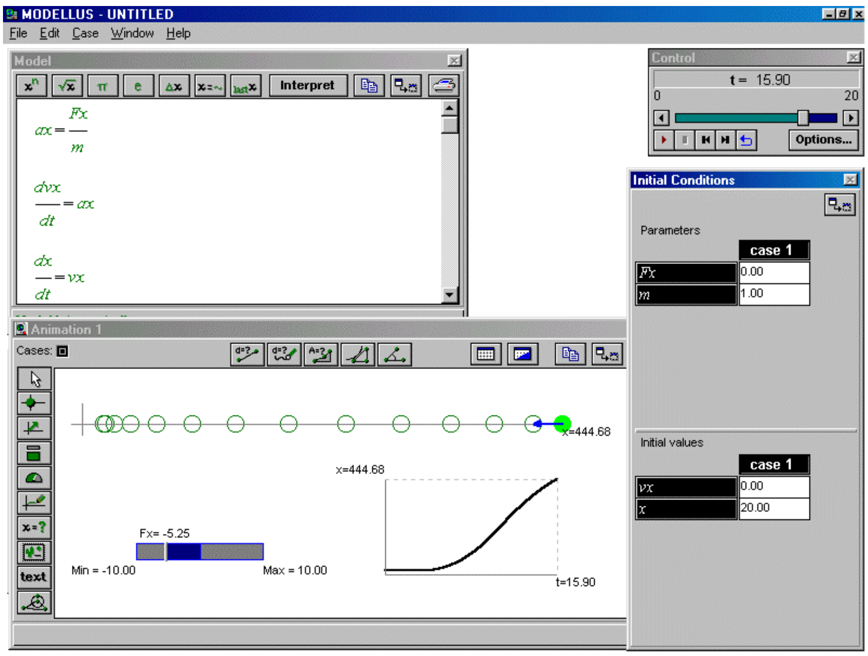


Figure 4.8: A 'Modellus' model of a particle moving on a straight line. The force on the particle, with an initial value of 0 N, can be controlled using the horizontal bar. In this example, after an initial interval of zero force, a force was applied to the right for a while (the particle increased its velocity), and then to the left (the particle decreased its

velocity). Changing the value on the bar changes F_x. Since v_x and x are computed by integration from F_x and a_x, its values are adjusted interactively.

The model starts with a simple computation to compute the horizontal component of the acceleration, a_x, from the horizontal component of the applied force, F_x, and from the mass of the particle, m. These two quantities are independent variables--it is necessary to give initial values for them on the *Initial Conditions Window*. Knowing a_x, Modellus can compute v_x by integration; again, knowing v_x, Modellus compute x, also by integration. These integrations are computed systematically at each time step. If F_x changes, all variables depending on F_x, either explicitly, like a_x, or implicitly, like the integrated variables v_x and x, also change at each time step.

Besides functions and differential equations, it is also possible to write another kind of models: models with difference equations or iterations. For example, an iterative model of decay of variable N using a difference equation. Iterative models are very useful particularly to study basic or advanced numerical methods, as I shall exemplify in the next chapter.

Despite more than a decade of criticism of exploratory environments based on programming metaphors, some authors (e.g. Eisenberg, 1995) still argue about the importance of programming in these kinds of environments. Programming is an evolving concept and it is now difficult to say if constructing a model with 'Modellus' is comparable with creating a program. Building a model with 'Modellus' involves certain characteristics of programming (e.g., there are syntax rules for building models) but, on the other side, many aspects of programming are "hidden" from the user (e.g., it is not necessary to create loops to assign different values for expressions). 'Modellus' has also some typical features of programming, such as conditions, particularly useful to define functions with different analytic expression for different parts of its domain.

With 'Modellus' it is also possible to analyse experimental data, either from photos or videos, or from any type of image such as screens from data logging software. For example, the user can take measures from a graph of experimental data, using 'Modellus' measuring tools (located at the top of the *Animation Window*). To make the model, the experimental data is placed as an image on the background of the *Animation Window*. The first step is, usually, to establish scales for the image (in the form of factor scales). Then, it is necessary to make convenient measures of specific parameters (e.g., inflexion points, period, amplitude, etc.). Finally, use the parameters to write the model and compare the model with the experimental data (taking factor scales into consideration) - see an example in Figure 4.9.

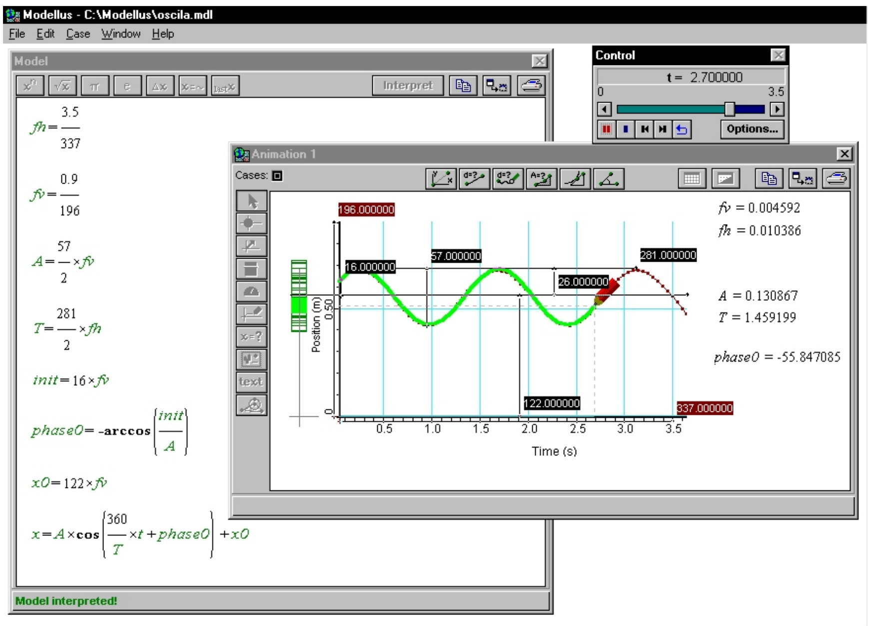


Figure 4.9: A 'Modellus' model made from experimental data--in this case, a graph of the coordinate of an oscillator obtained with data logging software. After obtained the model, this is compared with the data. The model was also used to make an animation of an oscillator - the small green square of the left of the graph.

A similar process can be used to photos (Figure 4.10) and videos (Figure 4.11):

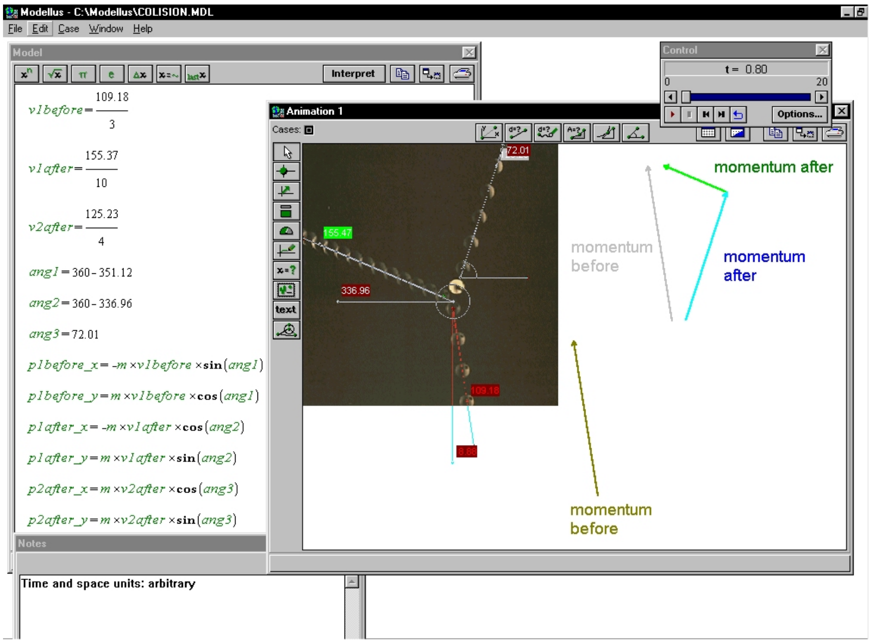


Figure 4.10: Using 'Modellus' to analyse a strobe photo of a collision. The measuring tools (for angles and distances) helped to find momentum before and after the collision. Momentum of each object is then represented as vector to check if there is conservation of momentum.

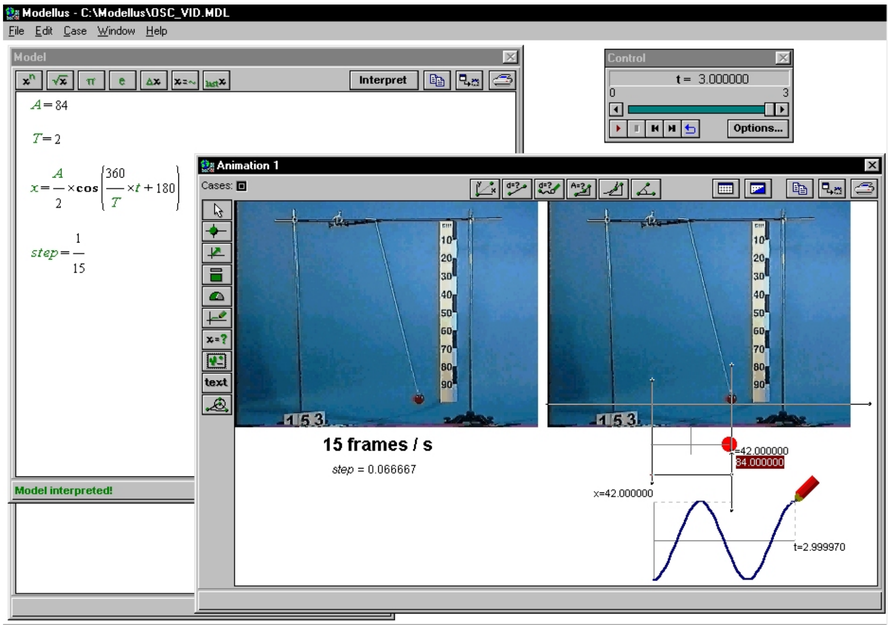


Figure 4.11: A model made from a video. After finding a time scale (15 frames per second), using a small step and counting images, it was necessary to measure the period and the amplitude of the pendulum. The model can easily be compared with the pendulum. The window shows two videos simultaneously. The left is the "original" video and the one on the right is a "copy" where the user can take measurements and overlap objects.

There are many other features of 'Modellus' not described here but documented on the 'Modellus User's Manual' (Knowledge Revolution, 1997; Teodoro et al., 2000).

4.2.3.7 Curriculum use of Modellus

'Modellus' has been used extensively in education in many countries. More specifically:

In the UK: It has been incorporated in the teaching of physics to post-16 students of Physics ('Advancing Physics' AS/A2 level - Ogborn and Whitehouse, 2000; 2001).

In Portugal: It is used in the teaching of Mathematics with pupils of 10-12 years old; its use is recommended in the official national curriculum. In Physics (8-12): There is some use of it with pupils of 8-12 years old; it is integrated in school books that its author (V. Teodoro) has co-authored. Overall, many teachers are familiar with the programme, and most of them are also using it.

In Brasil: It has hundreds of users in universities and schools.

In Greece, Poland, Slovakia: It has been translated by official educational authorities in the corresponding languages with a view to have it used by pupils in secondary schools.

In Netherlands: It is used commonly in schools and colleges.

In a number of other countries, including Germany, Chile, US, Taiwan, Colombia, Russia: there is some (but not much) use of it in schools and colleges.

4.2.3.8 Discussion of 'Modellus' from a pedagogical point of view

One of the most important features of 'Modellus is the possibility of exploring multiple representations of abstract objects. The concept of multiple representations has been a recurrent concept in exploratory software design for science and mathematics, at least since the publication of 'Making sense of the future' (Harvard Educational Technology Center, 1988). In this position paper, the authors argue about how computers can make a difference in learning environments. Besides other points, they stress the fact that computers can easily present simultaneously representations of the same object, such as a function (the analytical expression, a table of values, and a graph). Multiple representations, emphasizing different aspects of the same idea and affording different sort of analyses, are now a "take for granted" issue of most educational software for science and mathematics.

'Modellus' broadens the concept of multiple representations introducing the capability of creating visual representations of phenomena with a lesser degree of formalism than equations, tables, or graphs. For example, a traditional program for analysing functions, such as quadratic functions, permits the user to compare expressions, tables and graphs. But 'Modellus' also allows the user to create representations of phenomena where quadratic equations are used, such as launching a ball to the air. It also allows the user to explore and compare multiple contexts of the same phenomena, as can be seen in the above figures. However, this versatility of the programme comes with some disadvantages. It is tedious to set up a new animation, whereas it is easy to modify an existing one (Donnely, 1996).

Multiple representations in 'Modellus' are not limited to moving objects, graphs, equations, tables, etc.. It is also possible to create visual interactive representations of mathematical relationships, such as the one shown in Figure 4.12.

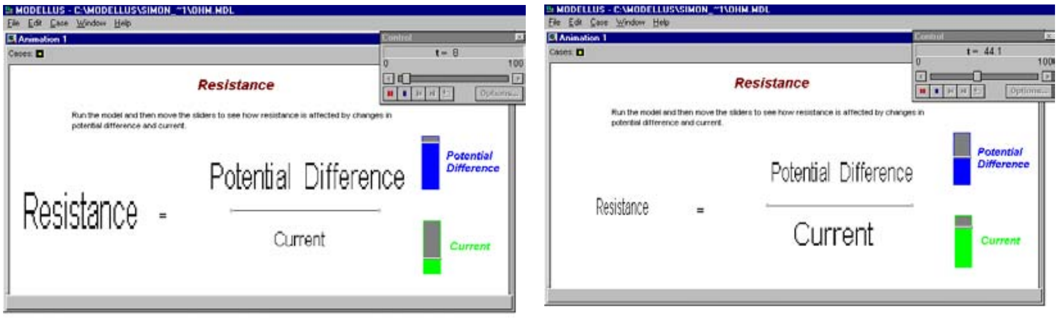


Figure 4.12: A visual explanation how change in the denominator and, or, on the numerator, affects the quotient (example made by Simon Carson, UK). Bars can be moved to change the "size" of the numerator and, or, the denominator, and seeing what happens to the "size" of the quotient...

From an educational point of view, 'Modellus' incorporates both expressive and exploratory modes of learning activities (Bliss and Ogborn, 1989). In an expressive learning activity, students can build their own models and create ways of representing them. In an exploratory mode, students can use models and representations made by others, analysing how different things relate to one another.

Teachers and curriculum developers can take advantage of the educational design of 'Modellus', since the software can be used as an authoring language for creating visual representations - see, e.g., Lawrence and Whitehouse (2000; 2001) or Fiolhais, Silva, Valadares, and Teodoro (1996a) each with hundreds of examples of visual illustrations of physics concepts done with 'Modellus'.

A teacher or curriculum developer can specify what is presented to the student in a certain learning situation. Since 'Modellus' windows can be hidden and files can be protected with passwords, a 'Modellus' example can show only what is appropriate for student's knowledge level. For example, Figure 4.13 shows a file where high school physics students can "play" with satellites without any knowledge of the differential equations that were used to create the model (the *Model* window is hidden and the file is protected by a password). When a file is protected by a password, buttons on the *Animation* window are limited to those strictly necessary, such as measuring tools.

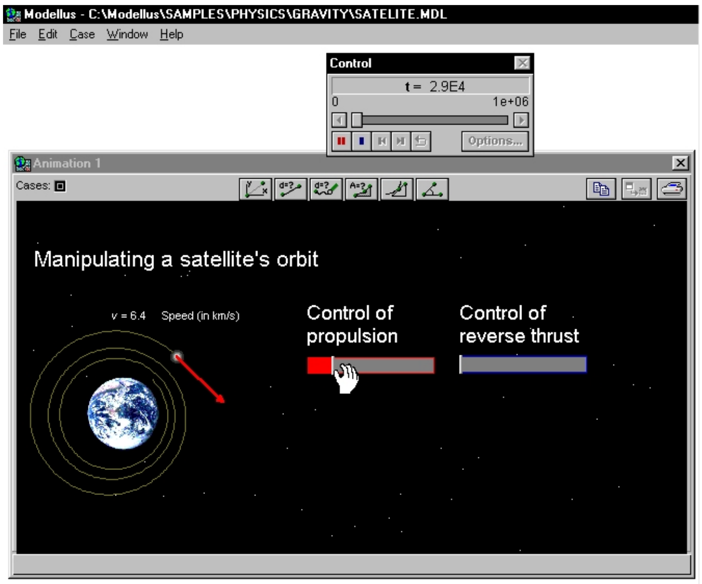


Figure 4.13: A 'Modellus' file protected by a password. This example allow students to explore satellite motion without any knowledge of the differential equations used to create the model.

The pedagogical design of 'Modellus' assumes that the computer is a cognitive tool, but not a replacement of the higher order human skills. Contrary to the design options of other educational systems, such as those usually adopted in intelligent tutoring systems, it is assumed that 'Modellus' is a tool to "impart wheels to the mind" but the intelligence, emotion, culture, poetry, and art, reside in the user, not on the software. 'Modellus' does not maintain any conversation with the user, or try to make any inference about the user's skills and purposes. It simply responds to user's actions. When using 'Modellus', it is assumed that conversations are activities made between students or between teachers and students, not between persons and software.

Learning with exploratory environments like 'Modellus' can never take place spontaneously: regulation and control are fundamental. Written materials, with guided inquiry approaches, that students read, discuss with peers, confront conceptions and descriptions, and where students must also draw sketches and write (the process of writing is a «disaccelerator» of information, specially visual information, and can act as an accelerator of knowledge construction) are essential approaches when designing learning environments with 'Modellus'.

'Modellus' can be used to help students understand scientific concepts in the sense given by Perkins (1993, p5), i.e., "performance understanding". A "performance perspective says that understanding a topic of study is a matter of being able to perform in a variety of thought-demanding ways with the topic, for instance to: explain, muster evidence, find examples, generalize, apply concepts, analogize, represent in a new way, and so on". Understanding scientific concepts with 'Modellus' must be doing things with them, not just repeating textbook explanations or writing equations out of any context. Performance-based understanding with 'Modellus' is simultaneously "hands-on" and "minds-on": "hands-on" in the sense that students can use concrete objects, like images, to make meaning of concepts. And "minds-on" in the sense that this use involves abstractions and its many representations. Interaction with 'Modellus' is engagement-based interaction--students must create and manipulate abstract objects. And, as Laurillard, says, "engagement leads to reflection; reflection leads to understanding" (1995, p. 180).

4.3 Analysis of modelling environments allowing creation of semi-quantitative models

4.3.1 Analysis of the IQON modelling system

For the needs of the research conducted by the 'Tools for Exploratory Learning' project* on the uses of modelling systems by students a semi-quantitative modelling tool - IQON (Interactive Quantities Omitting Numbers) – was developed.

IQON was based on BOX MODELLER (a prototype of a semi-quantitative modelling tool developed in Hypercard by Briggs (1989)) and was implemented in SMALLTALK.

The main idea of IQON is to provide a modelling environment *"to support the most elementary aspects of semi-quantitative reasoning and modelling, particularly to represent the idea that the qualitative magnitude of one variable (e.g. 'big', 'small' or 'normal') can change the value of another ('make it bigger', 'make it smaller')."* (Bliss, 1994b, p129)

IQON allows the user to represent a system in terms of interacting variables. Within the IQON environment it is not necessary to specify the values of quantities and analytical relations between variables. The user just has to define the variables, and to recognise changes and direction of changes between them by relating them in a diagrammatic way.

After constructing the model, the user can run it as a simulation. At that time, the system 'assumes the control of the model', and automatically attaches to it a set of dynamic mathematical equations (hidden from the user), which govern the simulation. Following this, it makes the model iterate over time, giving visual feedback on the screen by changing the levels of the appropriate variables.

The next two subsections are based on Sampaio's PhD thesis (1996). They present the IQON interface and its use in more detail, and offer a critique of the software. The version of IQON discussed is a prototype developed for Macintosh computers using system 6.

4.3.1.1 IQON: General Description

IQON uses a direct manipulation interface to implement and interpret causal-loop diagrams in a consistent way. When the system is loaded it presents the user with a window where models can be constructed or loaded (see Figure 4.14). At the top of this window there is a control panel with 13 icons that are for: (i) creating and manipulating the building blocks on the screen (9 first icons); (ii) running a model (3 icons); and file manipulation (1 icon).

* The 'Tools for Exploratory Learning' programme was run by J. Bliss, J. Ogborn, J. Briggs, D. Brough and H. Mellar between 1989 and 1992. The programme was funded within the ESRC's Initiative on Information Technology in Education.

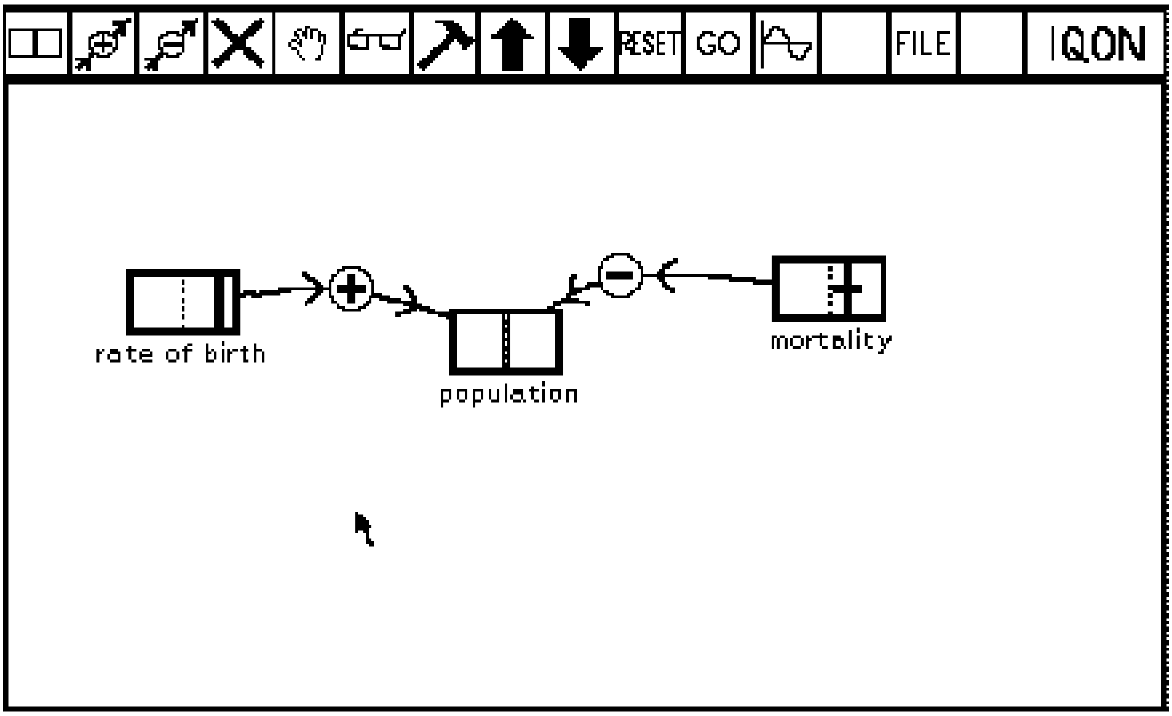


Figure 4.14: IQON screen with a simple model about Population

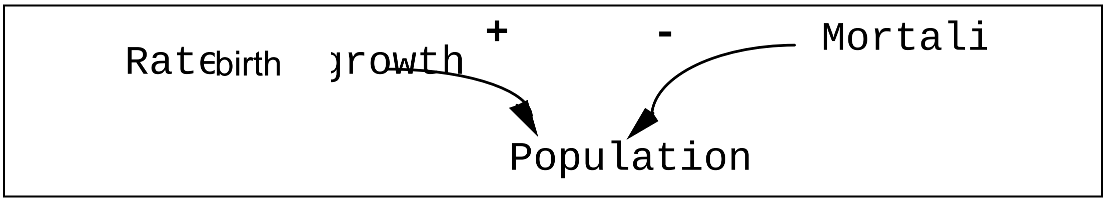
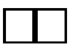


Figure 4.15: The causal diagram which corresponds to the model shown in Fig. 4.14

The system has two kinds of primitives (or building blocks): a *continuous-valued* variable (also called "box") that can assume semi-quantitative values compared to a *normal* level; and *positively* and *negatively affects* links to represent incremental changes between variables.

To create an object or to perform some operations on the objects on the screen, the user has to "press" an appropriate button on the control panel. Once an icon is selected, the system enters in a mode of operation associated with that icon. To change the mode, the user has to click on another icon (which will select another mode) or in the case where the system is in a 'go'/'reset'/'graph' mode, the user can click anywhere on the working area to leave that mode.

Presented below is a description of the main buttons and the operations associated with them (see Figure 4.14).

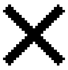


The 'box' button permits the user to create a continuous variable on the screen. All variables have to have a name which has to be given before the variable is shown on the screen.

A continuous variable contains a level bar that can be moved -either by the user or by the system when running the model - to the left or to the right. When it is in the middle of the box it is said to be in a resting state (indicated by a dotted line) and does not exert any influence on the dependent variables. The greater the deviation of the level bar to the right the bigger the value represented by the box. It is the same the other way around: the greater the deviation of level bar to the left the smaller the value represented by the box. The response of a level is non-linear, limited by the ends of a box, which represent values as far above or below the 'normal' level as one cares to imagine.

Links are used to represent causal relations between variables and can pass positive or negative influences between variables. The way they affect a certain variable (by incrementing or decrementing its level bar) is determined by their sign: In the example shown in Figure 4.14 *rate of birth* positively affects *population* and because *rate of birth* is high then it will push up *population*. However if it were the case that *rate of birth* was low then population would be pushed down. A negative link works the other way: In the same example *mortality* negatively affects population and because it is high then it will push down population. If it were the case of *mortality* being low then it would push up *population.*

Links can also have different relative strengths implying a strong effect between a certain variable onto another. The system represents this idea by using bold '+' and '-' signs.

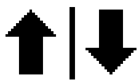


The 'cross' icon is used to delete variables and/or links. If a dependent variable is destroyed all of its incoming links are also automatically erased by the system.

The 'hand' button serves to reposition boxes and links on the screen. After selecting it on the control panel, the user has to click and drag the object he/she wants to move.

The 'spectacles' icon is used to give access to the internal parameters of a variable (its name and comments about it) also permitting the user to change these parameters.

The 'hammer' button is used to fix/liberate the level bar of a variable. When the level bar of a variable is fixed, its value does not change during a simulation. If the variable is a dependent one, all its links are redrawn using a shaded pattern to indicate that they will not work (as if they were "sleeping') during the simulation.



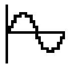
The 'up' and 'down' arrows permit the user to increase and decrease the level bar of the variables shown on the screen. After selecting the arrow on the control panel, the user has to select the variable he/she wants to change by clicking on it. Each new click on a variable will make its level bar move up or down. These arrows can also be used to change the sign of links. The up arrow will make negative links become positive and the down arrow will make positive links become negative.

GO

By pressing the 'GO' button the model is made to run. When it enters this state the system automatically generates a quantitative model (a set of algebraic or differential equations always hidden from the user) based on the diagram on the screen. During the running process, each variable averages its input (In case it has more than one input) taking into account the relevant signs of the ingoing links to calculate its value for the next iteration. To stop running a model, the user can click anywhere on the screen.

RESET

The 'reset' button resets the system clock (which does not appear on the screen) and makes all variables go back to normal position (internal value = 0).

 The graph button permits the user to visualise the variation of a certain variable (chosen by clicking on the desired variable) against time. The graph is not constructed in real time and only the result of simulation for one variable at each time is available (see Figure 4.16).

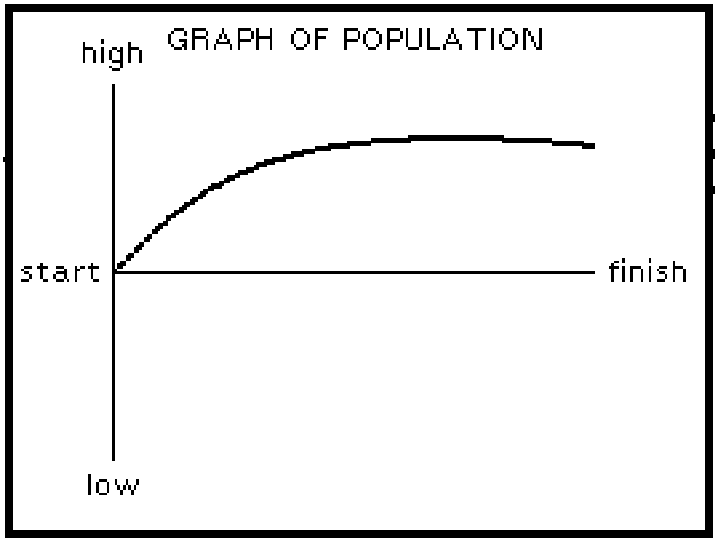


Figure 4.16: Graph of the variable Population X Time for model shown in Fig 4.14

4.3.1.2 Mathematics of IQON System

In IQON the mathematics that controls the behaviour of the variables during a simulation is always hidden from the user. He/she is always thinking about both the model structure and its behaviour purely in terms of qualitative and semi-quantitative ideas. According to the developers of the system (Miller *et al.*, 1990) its mathematics was partially inspired by looking at the mathematics used in developing connectionist nets (Rumelhart and McLelland, 1987). The properties of those systems used in IQON are: (i) they have a resting level; (ii) they respond over a limited range of values so that they can be interpreted as big or small values; (iii) they respond to inputs in a simple and uniform way by taking a weighted sum of them[12] (Miller and Ogborn, 1993). The Figure 4.17 below shows the correspondence between an IQON model and the mathematics that control its behaviour.

[12] A complete description of IQON's mathematics is presented in Miller and Ogborn (1993).

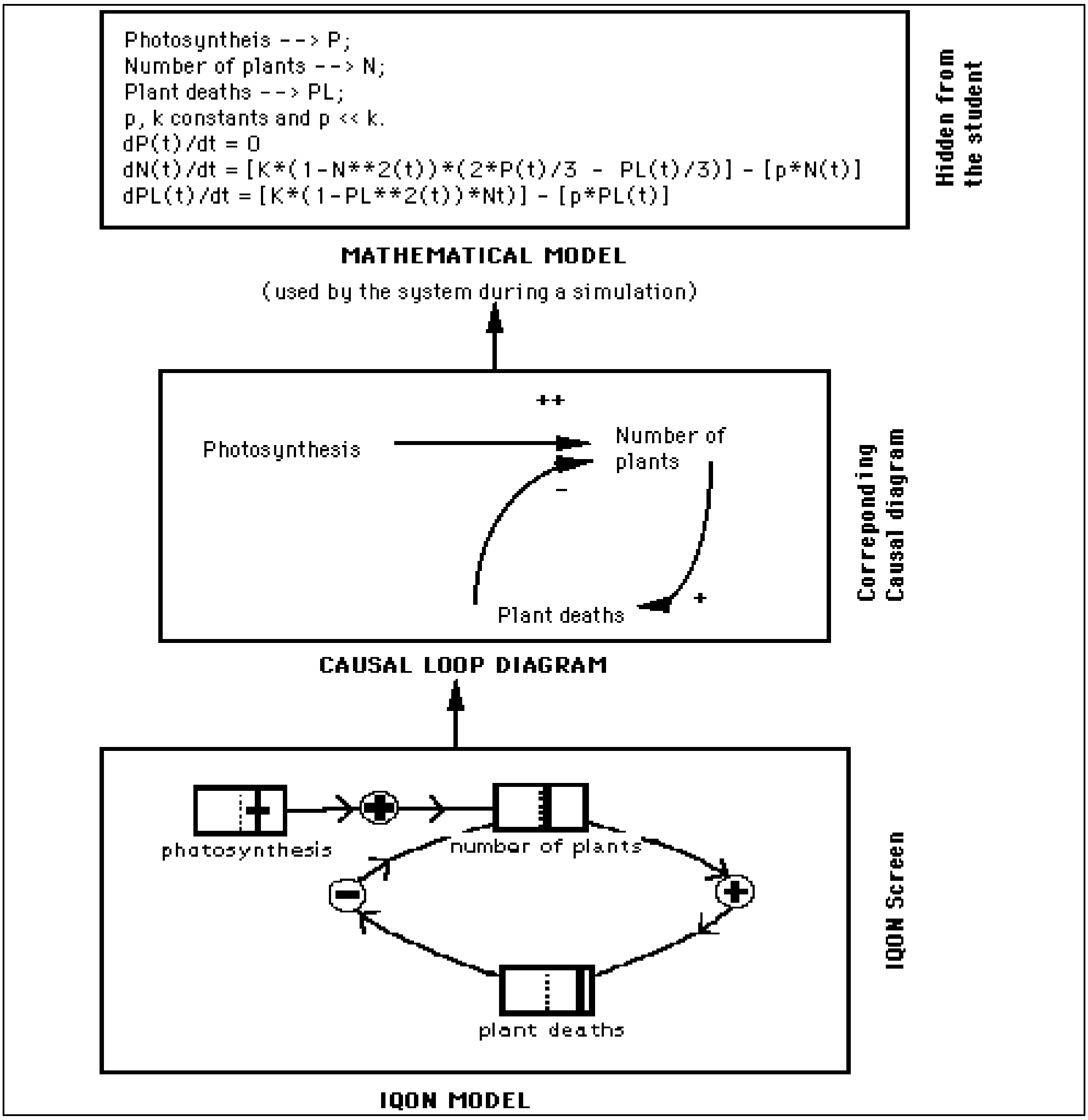


Figure 4.17: Correspondence between an IQON model, causal loop diagram and the underlying mathematical model

4.3.1.3 Discussion of IQON System

It is well known that when someone is trying to understand a situation, he/she normally has too little knowledge to form an exact quantitative model (Dillon, 1994). So, starting work with quantitative modelling to think about the dynamic behaviour of systems does not seem to be the appropriate way because in such situations the user is obliged to focus on the functional behaviour of the variables before having a systemic view of the problem. IQON came exactly to fill this gap by opening up the possibility for the students to create models without having to be mathematically precise about the relations between their constituent elements (Ogborn, 1990).

Another important aspect of such a system is that it permits younger students (12-15 years old) to be initiated into modelling activity. The reason is the same as that stated above: there is no need to know formal mathematics in order to use the system (Bliss *et al.*, 1992).

The following two subsections discuss the aspects related to the interface of the software and its modelling possibilities respectively.

IQON: Discussing its Interface

A very positive aspect of the system is its simplicity for qualitatively representing a certain phenomenon. The metaphor of *boxes* and *arrows* is very powerful to represent the idea of values and relationships. In a certain way we are accustomed to seeing this kind of diagrammatic representation in different sorts of printed material such as books, newspapers and folders. It can also be easily mapped onto the idea of causal loop diagrams with *boxes* representing system variables and *arrows* the relationships between them.

Compared to STELLA's metaphor of *stocks* and *pipes* the one used by IQON is simpler and yet more general (it can be seen as a superset of the metaphor used by STELLA). A good example is to compare the representation of physical entities such as *velocity* and *displacement.* With IQON the level bar just moving across the box can give the user the idea of something increasing or decreasing instead of filling or emptying a container. When it is the case of modelling a variable that accumulates and/or de-accumulates the visual effect of the level bar combined with the variable-box can also be easily understood as a container being filled or emptied.

Another important aspect of IQON's metaphor is that it permits the visual representation of negative values as something 'being below the normal level'. STELLA again, as mentioned above, does not visually provide the user with the idea of negative numbers.

On the other hand, a negative aspect of the interface is that it requires excessive mouse operations to perform certain interface tasks. For example, when the user wants to move the level bar up or down, instead of directly going to the level, clicking on it and dragging it to the right or to the left, the user has first to select the appropriate button on the control panel ('up' or 'down' arrow) and then click on the desired variable as many times as the number of increments/decrements he wants. Worse still, if in the middle of this operation the user wants to move the level in the contrary direction, he/she has to click again on the other 'arrow' button on the control panel and repeat again the process of clicking on the variable.

Although the system provides the user with a plotted graphic as another way to represent the output of a variable, it is nevertheless very limited. The graph only shows the variation of the variable against time for a certain pre-determined period of time which is not known by the user. The graph is not constructed in real time and can only be presented after a simulation has been tried.

Another negative aspect is that the graph of any variable presents the internal value of the variable, using the hidden damping function (employed by the system as a relaxation function to make the system stable) as part of the calculation of the variable. So if the user defines that *population* increases with *rate of birth* and he/she only sees the graph of *population* versus *time* (Compare the model shown in Figure 4.14 with the graph output shown in Figure 4.16), he/she will have the wrong impression that *population* stops increasing after a certain period of time.

IQON: Discussing its Modelling Possibilities

Although IQON permits the user to model a wide range of problems related to the secondary school and undergraduate curriculum (Kurtz dos Santos, 1992), there are particular examples of these problems that are not well handled by the system:

• Conditional events - The occurrence of some events is sometimes conditioned by other events. Mathematical formalism permits to represent these situations by using constraints within a set of equations. A typical example is the fact that certain ecosystems like a lake can adapt themselves to the release of pollutants up to a certain level. If the concentration of the pollutant in the water becomes greater than a certain value, the animals and plants that are dependent on that environment begin to die or to have severe problems[13].

• Direct relationships - Not all relations in a dynamic system are the rate of change type. Actually there exist some relations that are functional and can be represented by algebraic equations. A typical case is the relation between the physical entities *force* and *acceleration.* As it is well known in dynamics they are associated by the equation F = m * a.

• Combination of inputs - There are many problems in nature that are not appropriately modelled by using the 'average' combination provided in IQON. A typical example is rust. It is a known fact that in order for rust to occur (for instance in a car) two components are necessary: water and oxygen. If in a certain environment only one of these components exist then there will be no rust. The present input combination used by IQON is not appropriate to model such situation because it averages the inputs of a certain variable no matter what their values. Another situation would be when the user wants to add the causal factors to calculate the value of a dependent variable. In the example given above about lake pollution, it is evident that if more than one pollutant is being released in the lake the elapsed time until the lake becomes "polluted" would be proportional to the sum of the pollutants and not to their average[14] .

• 'Positive/negative only' variables - Variables that have values that can be above or below a certain level are not the rule for all problems in nature. The example above about the lake pollution serves as a good example here as well. There is no sense to the idea that a lake is polluted 'below the normal'. We know that either a lake is not polluted or it is polluted at a certain level. In this example the normal level should be interpreted as the smallest possible value for the pollution of the lake. This is the same case for rust (if someone is analysing the problem from a chemical point of view). Either it exists or not. There is no "real world interpretation" for rust being 'below normal' (or less than zero). The system should provide a way for the user to say so.

4.3.1.4 Pedagogic Use

Research was carried out in the Tools for Exploratory Learning Programme, using IQON and looking at how pupils between the ages of 11 and 14, reasoned when working with such a semi-quantitative tool.

The results of the study suggest that pupils of those ages can use the tool to create their own models. *"They can understand that a model is fallible and can be modified or that they may need to build a new one."* (Bliss, 1994b) Moreover, it seems that IQON provides the opportunities to pupils to learn more about some fundamental aspects of modelling, such as the relation of models to reality or about how a model (or modelling system) looks at the world in its own special way.

[13] The idea of simulating conditional events is the focus of Discrete Event simulations. Within this category the modelling primitives include the notions of events and delays instead of variable-type primitives.

[14] It is also true that the time is also related to the type of pollutants. But even if someone makes an approximation considering that the pollutants have the same "power of pollution", IQON can present very unexpected results for certain combinations of inputs.

4.3.2 Analysis of WlinkIt system

WLinkIt is a computer based modelling system which also belongs to the category called *semi-quantitative* systems. The system can be used to represent causal relationships between objects and/or events of a certain domain. The vocabulary used in its environment tries to correspond well to intuitive or common-sense ways of talking about systems.

WlinkIt is the Windows version of the second version of LinkIt (Sampaio, 1996; Sampaio and Ogborn, 1996; Sampaio, Santos and Ferracioli, 1999), originally developed in cT language in 1996 by Fabio Sampaio.

4.3.2.1 General description of WlinkIt

The construction of a model with WLinkIt is made through the use of a direct-manipulation interface using a graph metaphor to define the objects and relationship of a certain model. In most cases these objects can be used by the modeller to represent some aspects of the world to be modelled such as the level of pollution of a city, its main causes and its consequence on the health of the population of that city (See Figure 4.18).

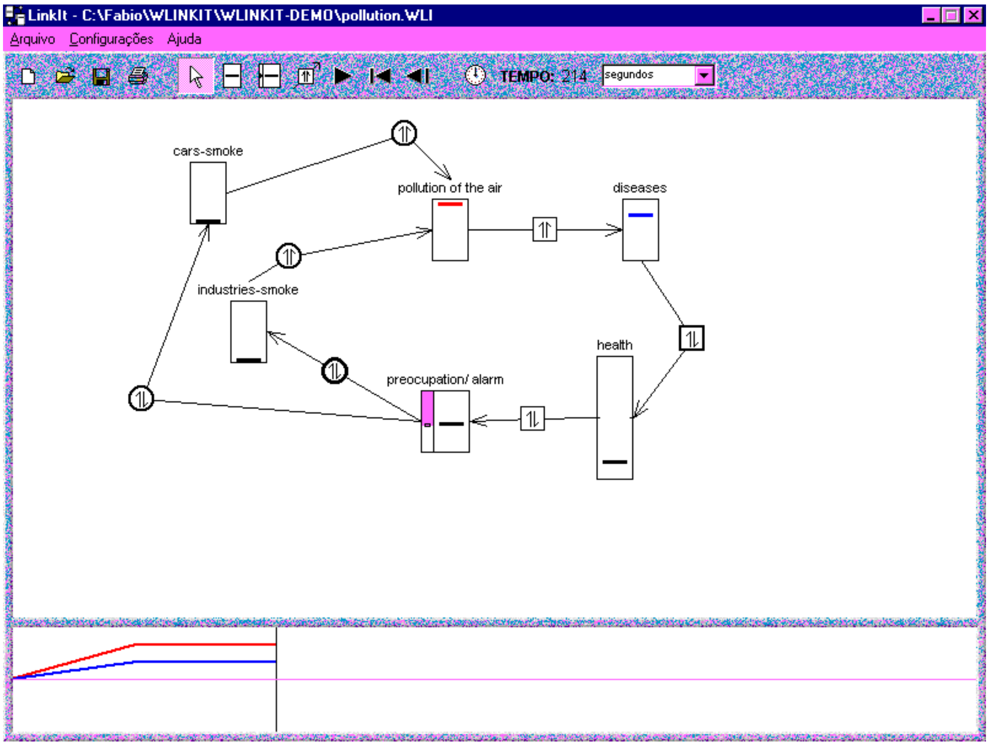


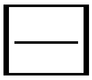
Figure 4.18: LinkIt's window with a model being constructed (the graph shows the variables *pollution of the air* and *diseases*)

To create a model with WLinkIt, the user has to define the variables and relationships and attach to them some characteristics that reflect the functional aspects of the objects and events being modelled. The system, in its turn, is responsible for running the model. In order to do so, it executes two steps: First it has to choose a set of mathematical equations (always hidden from the user) - based on the characteristics of the variables and relationships already created - that will control the behaviour of the model. Second, it makes the model evolve over time through the calculation and iteration of these equations. During this process of running the model, the user receives visual feedback from the system, seeing changes in some physical aspects of the objects on the screen.

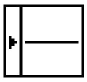
4.3.2.2 WLinkIt components

WLinkIt has two main system-objects: **variable** and **link**.

Object-variables are presented to the user as a box (called variable-box) containing one or two level bars inside it. This type of object can be of two forms:



Smooth variables can be used to represent any factor about a problem someone wants to model. They are shown on the screen as a rectangular box with a horizontal amount level indicator in the middle (e.g., variable *pollution* in Figure 4.18). They can be of two different range: [0, +1] (e.g., *pollution*) and [-1, +1] (e.g., *health*). These two ranges combined with the visualization of the amount level on the screen, permit the representation of positive and negative values or more qualitative ideas such as above or below the "normal" level.

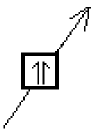


On/Off variables can be seen as a special case of *Smooth* variables. They serve to represent conditional factors that can control the behaviour of dependent variables. Their box is divided into two parts. The left part contains a threshold level indicator (a small triangle) used to define when the variable will trigger and the other part of the box contains the amount level indicator. Both of them can be moved by the user independently. (e.g., variable *preoccupation/alarm* in Figure 4.18).

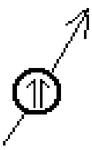
When a variable is created its default range is set to 'positive only' values. If the user changes it to 'any value' (through the *Information* box of the variable), the height of the variable box doubles (e.g., variable *health* in Figure 4.18) and all links that arrive or depart from it are redrawn to fit the new size of the box.

The resting level (internal value = 0) of an 'any value' variable is indicated by two small dots located on the left and right side of the box.

Links are used to connect variables and define a causal relationship between them. Basically the system provides two different types of links:



Go together links are used to define relationships where the value of the affected factor is immediately calculated based on the value of the causal factor. Mathematically it serves to represent relationships of the kind *y=a*x*, where *a* is a constant that can be modified by the user. The value of *a* is visually represented in the model by the sign showed inside the square box in the middle of a link (e.g., according to Figure 4.18 *diseases=+1*pollution* and *health=–1*diseases*).



Gradual (or *cumulative*) links are used to define relationships where the value of the causal factor can be seen as a rate of change of the dependent factor. Mathematically it serves to represent relationships of the kind *y(t+1)=y(t)+a*x*, which is a discrete time step approximation of the linear differential equation *dy/dt=a*x*, where *a* is a constant that can be modified by the user. The value of *a* is visually represented in the model by the sign showed inside the circle box in the middle of a link (e.g. according to Figure 4.18 *pollution(t+1)=pollution(t)+1*car-smoke* and *industries-smoke(t+1)=industries-smoke(t)–2*preoccupation-alarm*)

Also the links that arrive at a certain variable (called input combination) can be combined in one of three different forms: they can be summed up (or subtracted if an opposite link is used), multiplied (or divided if an opposite link is used) and averaged.

All of these different types of variables with their different input combinations and links with their different direction of effect and strength permits the system to model a wide range of problems that are part of the school science curriculum.

4.3.2.3 Modelling possibilities and limitations - Examples

WlinkIt can be used to construct models which belong to a number of classes of mathematical models: algebraic models, boolean models (with the aid of the *On/Off variable*) and differential equations.

More specifically, Table 4.1 summarises the different possibilities of differential equations that can be modelled with WlinkIt.

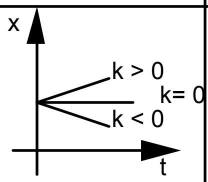
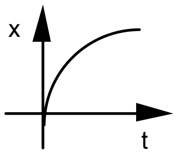
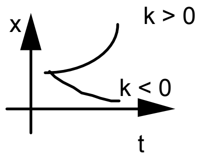
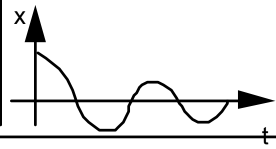
	Linear processes	Build up exponential	Exponential growth & decay	Oscillations
Ex. of problems	Pollution Population Fluids Eletromagnetic induction Etc.	Leaky tank Pollution LR circuit Etc.	Population Food absortion Nuclear decay RC circuits Etc.	Spring-mass system Pendulum Predator-prey Etc.
Dif. Equations	$dx/dt = \pm k$	$dx/dt = K*(M - x)$	$dx/dt = \pm Kx$	$m*\frac{d^2x}{dt^2} + b*\frac{dx}{dt} + kx = 0$
Order	1	1	1	2
Possible graphical solutions				

Table 4.1: A summary of dynamic modelling possibilities of WLinkIt

Simple algebraic models, however, which do not include differential equations, might be better represented as systems of constraints, where one could calculate any quantity given the others. This is not possible in WlinkIt, which follows the direction of links in calculating quantities. Moreover, loops of links in systems of algebraic equations might sometimes give problems, since they are not disallowed by the programme (Ogborn, 1999).

4.3.2.4 Research results

LinkIt (the first version of WlinkIt) has been used in Brazil with pupils of 13 to 18 years (Sampaio, 1996; Ogborn, 1999). Its use has demonstrated that pupils of these ages can easily learn the basics of the system and can construct models of their own. However, according to the researchers and developers of the system (e.g. Ogborn, 1999, p37), the most valuable pupil activity that LinkIt produces is not the models made or seen, but rather the talk and discussion that accompany their making and use.

Concerning the functionalities of the system its use has shown that:

- Pupils had some difficulty differentiating between an inverse link (corresponding to 1/x) and a subtractive link (corresponding to –x).
- It was also not clear to them why a subtractive link of a negative value has a positive (i.e. opposite from what they may expect) effect on the dependent variable. *"Indeed much of their thinking about variables and the relationships among them is 'positive thinking': They prefer 'same direction' links to 'opposite links' where possible, even if it means inverting the name and meaning of a variable."* (Ogborn, 1999, p36).

4.3.3 Analysis of Model-It system

4.3.3.1 Historical data

A software first called ScienceWorks Modeler and later Model-It was developed at the University of Michigan (USA) as part of a project integrating computing technologies into the daily life activities of students (Jackson, Hu and Soloway, 1994; Jackson, Stratford, Krajcik and Soloway, 1995). The software was developed originally in Smalltalk for Macintoshes only. The discussion below is based on version 3.0 of Model-It (7/11/01).

4.3.3.2 MODEL-IT overview

Model-It provides a direct manipulation interface to build a wide range of process flow models. To construct a model pupils have to identify a set of objects related to the problem to be modelled and define variables associated with these objects. The objects in a model are displayed using photo-realistic graphics that students can create themselves or can select from the set already provided (see Figure 4.19). All Objects must be created while in the Plan mode. There are two possible types of Objects: Normal and Background. Each model contains only one Background Object. Every object has four characteristics: Name, Type, Image, and a Description. The Background object provides context for the model.

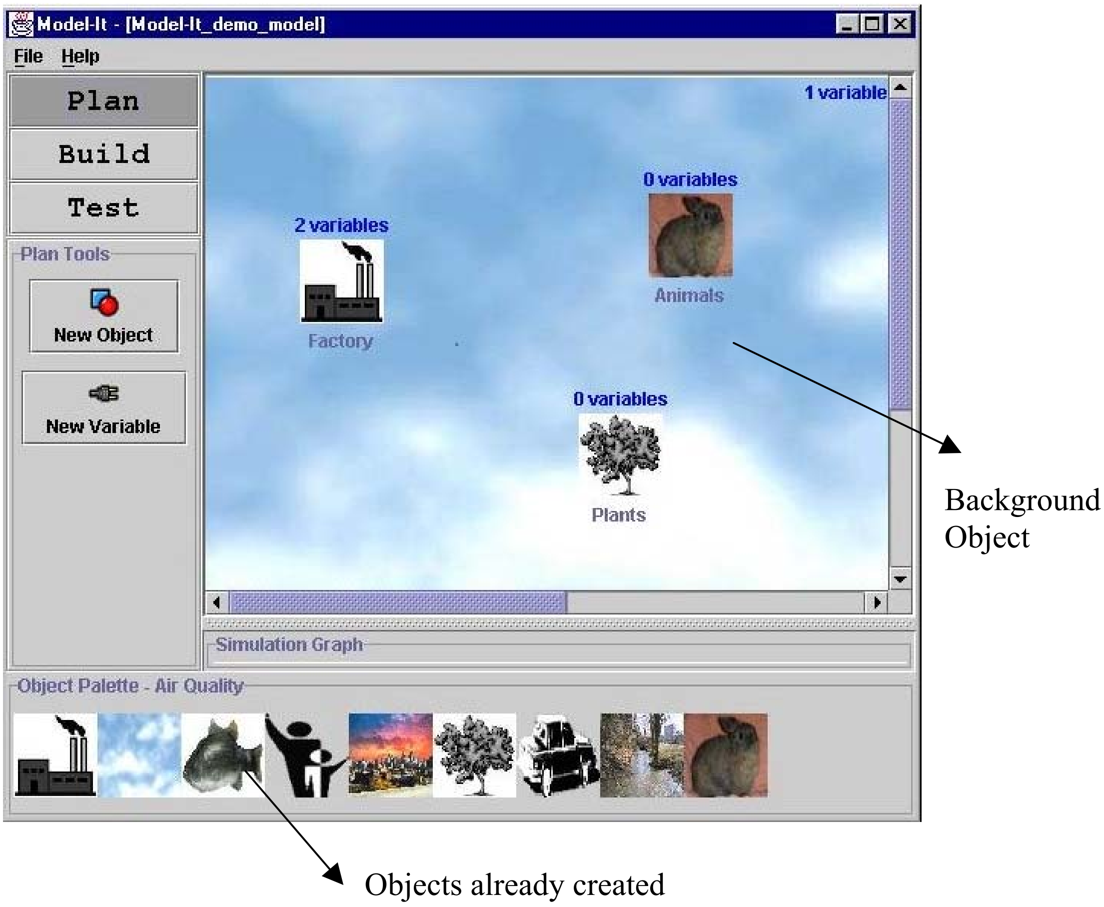
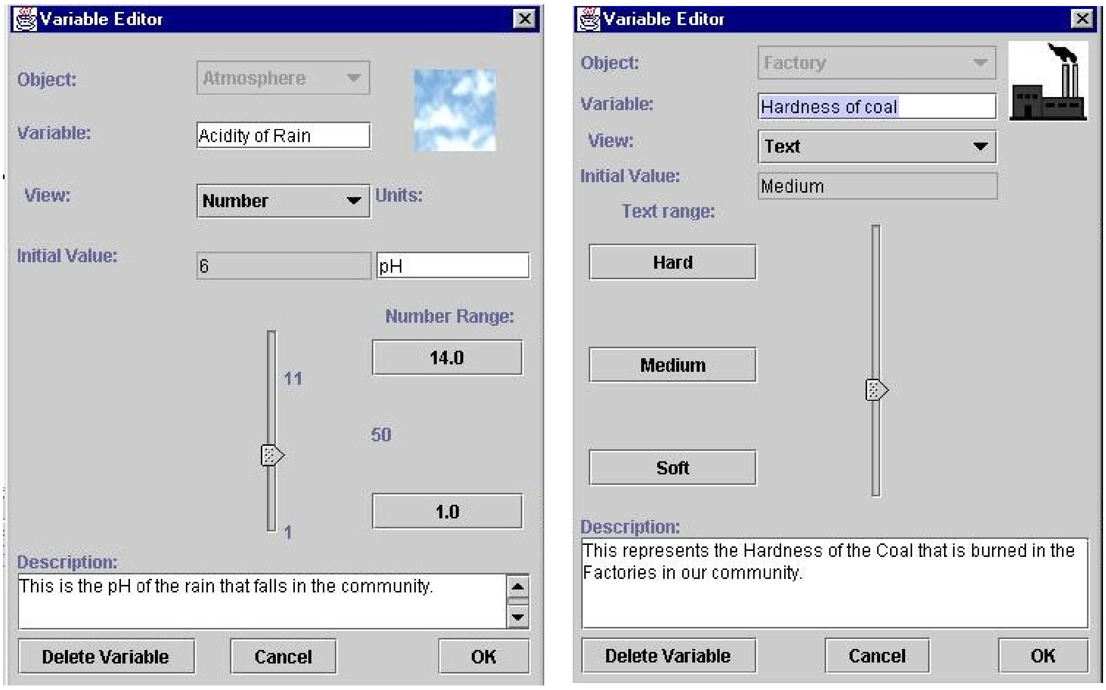


Figure 4.19: Main window of Model-It for model 'Air quality' showing a digitized image of clouds to represent the Background object 'Atmosphere', as well as three other objects related to the model.

All objects need to have measurable traits. These measurable traits are called Variables. To define the variables associated with these objects the user has to call the 'Variable editor'. Variables can take values which are either numerical (Figure 4.20a) or are characterised by some text description (e.g. high, medium, low) (Figure 4.20b).



(a) (b)

Figure 4.20: Model-It - Variable Editor

Then, pupils have to build relationship links between the variables, using a variety of representations from qualitative to quantitative. A graphical view of each relationship is also provided.

Relationships are build in the Build model. The independent variable is called 'causal' and the dependent one 'effected'. A relationship is defined in the Relationship Editor (Figure 4.21). The definition of a relationship is made through a qualitative verbal representation by selecting descriptors in a sentence. The pupil is prompted to justify in writing the descriptors s/he chooses. The system basically permits two types of relationships: absolute (i.e. linear, exponential or a 'bell curve') and rate of change (see discussion in following section) and permits two directions of change: 'increases' and 'decreases'. The Relationship Editor also helps determine if View by Text or View by Table is appropriate.

Once relationships are built students can test their models to see the results. Testing the model involves manipulating independent variables to see how dependent variables are affected. For data visualization Model-It provides meters and graphs to view the values of variables. While a model is running students can change the values of variables and immediately see the effects. (Figure 4.22)

The designers themselves note that *"Model-It can not determine if a relationship is appropriate or not. Only through testing and class discussion can such determinations be made"*.

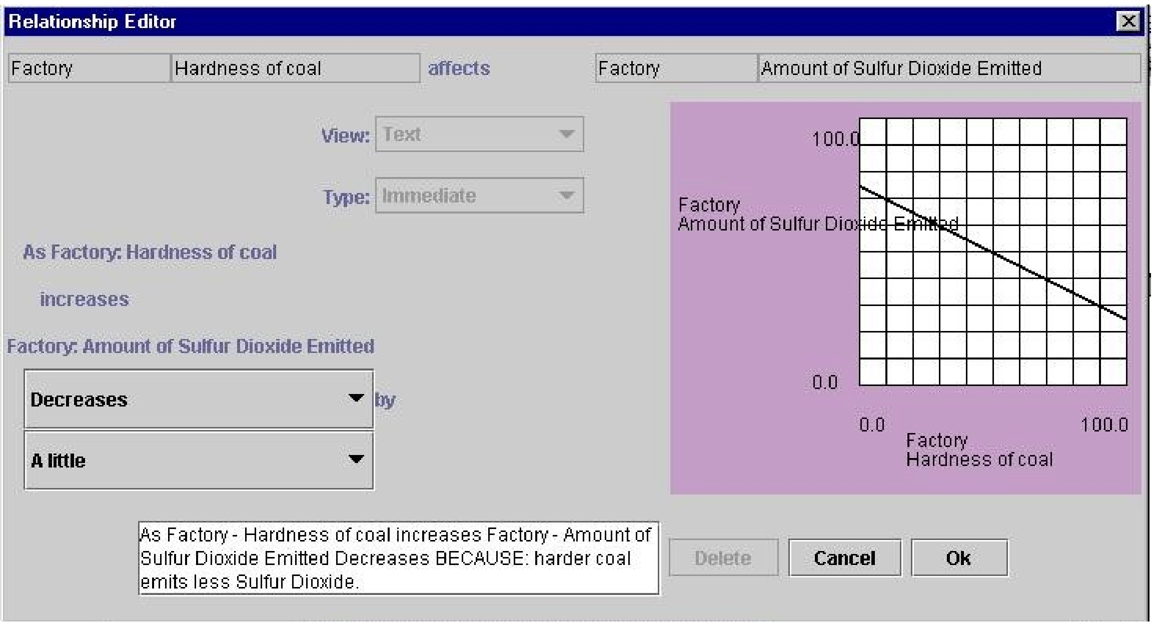


Figure 4.21: Model-It – Relationship Editor

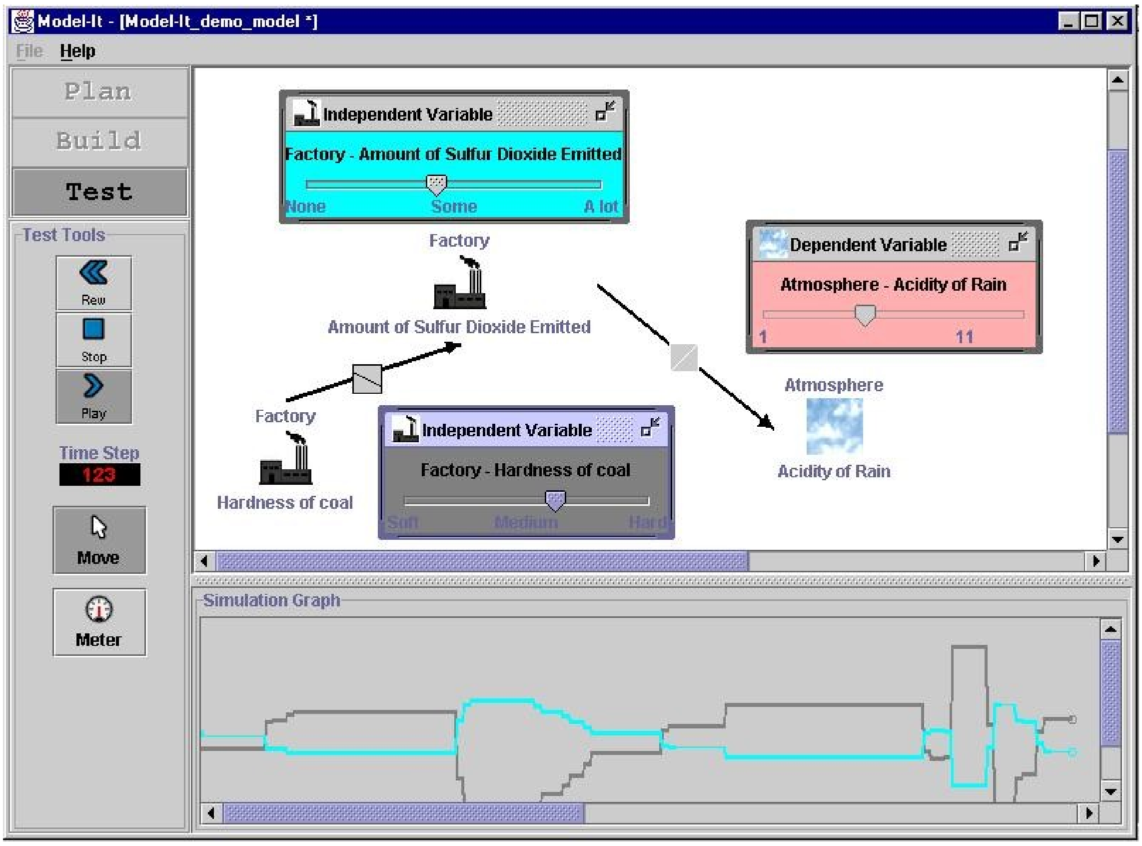


Figure 4.22: Testing a model in Model-It

4.3.3.3 Underlying representation system

Model-IT supports structural and behavioural representation of physical systems. The structural representation system is a simple object-oriented system. The system is simple because it does not support inheritage specification, hierarchical organization of objects and definition of methods. The behavioural representation system is based on predefined mathematical equations that the user can select to specify the kind of relationship between object variables. This results in a system of mathematical equations (simple and differential) between object variables (quantitative variables organized in containers-objects).

The equation set solution is implemented in Model-IT by a specific algorithm that propagates change among the variables. The execution is not absolutely mathematical accurate but produces realistic behaviour in most cases.

The structural representation system

The conceptual structure is a high-level data model of the system under study. A conceptual structure in Model-IT is a set of objects. An object is defined by a name and a list of properties ('variables' according to Model-IT terminology). A variable is a data container of a simple type. That is Model-IT enables the user to specify problem domain concepts organised in a simple object-oriented data model.

The functional (behavioural) representation system

A Model-IT model constituted only by objects and variables cannot be executed (simulated). That is objects and variables represent only the informational structure of a system. In order for a Model-IT model to be executable the user must define the control structure, which reflects the functional characteristics of the modelled system. In Model-IT relations between object variables specify the control-structure of models. The relations are quantitative but specified roughly by selection from pre-defined categories (semi quantitative modelling). A relation can be 'Immediate' or 'Rate'.

Immediate relationships

Immediate relationships are specified by means of a correspondence table or using phrases of the following syntax:

Object1.variable **affects** *object2.* variable according to the following relationship:

As *Object1.variable* [**increases** or **decreases**] *object2.variable* [**increases** or **decreases**] by [**a lot** or **a little** or **more and more** or **less and less** or **a bell-shaped curve**]

The semantics of immediate relationships are of the form y=f(x); the value of the 'affected' variable is immediately calculated using the value of the 'causal' variable.

A model with relationships only of immediate type is equivalent to a system of mathematic relations. The values of the depended variables are calculated by the values of the independent variables. (There is no independent time variable). In order for the model to get into a new state an independent variable must take a new value. The dependent variables take their new values as soon as the first change occurs. This kind of models could be used to explore simple modelling concepts such as chains of relationships (Jackson, Stratford, Krajcik and Soloway, 1995).

Rate relationships

Rate relationships are discrete time step approximation [$y(t+1)=y(t) \pm x$] of the linear differential equation $dy/dt=\pm x$. For population objects the above equations becomes $dy/dt=\pm x*n$ where n is the previous number of population members. The rate relationships permits the exploration of dynamic time-based systems furthermore the latest relationships permit to model exponential growth of variables.

The model execution algorithm

According to Jackson, Stratford, Krajcik and Soloway (1995), in Model-IT prior to a model's simulation every qualitative definition is converted into quantitative functions. (In Bobrow, D. G. (1984) the interested reader could find ideas about qualitative reasoning and systems simulation using AI techniques.) The text-based immediate relationships are converted into the functions presented by their associated graphs, and scaled to the defined range of values for each variable. In the case of definition of relationships via the correspondence table the value pairs are stored and linear interpolation is used for non-specified value approximation. Objects maintain lists of their associated variables, and variables keep track of their initial value and current value. Variables also keep track of which relationships they cause.

For each time step in simulation the Modeller's central Controller cycles through a loop step and executes two functions:

(1) Firing relationships. All defined relationships are fired. When a relationship is fired, it calculates a new value for the affected variable according to the equations described above, and tells the affected variable its new value. The affected variable stores the new value in a list.

(2) Calculating new values. Once all relationships have been fired, the variables calculate their new value, if any, from the stored list of new values. It is possible that more than one value may exist for a variable, if this is affected by many other variables. The new value is determined according to the kind of relationships (immediate or rate): For immediate relationships the new value is the average of the multiple values. For rate relationships the new variable value is computed by the addition or subtraction of all new values to the old.

The mathematical semantics of the above algorithm and its consequences on the accuracy of the simulation are debatable, however accuracy is not the primary goal in semi-quantitative modelling.

Summary

Summarizing Model-IT uses a simple object-oriented language for the conceptual modelling of physical systems and semi-quantitative semantics for the functional representation. In order for the system to produce simulations a mapping has been defined between the roughly specified relationships and mathematic equations. This mapping is rather accurate but this is not a problem for the purposes of Model-IT as far as it does not become the cause of misconceptions.

What systems' domain does the tool properly affront?

Model-IT concerns mainly dynamic systems modelling like STELLA, POWERSIM etc. The main differences are the specification of qualitative relations and the grouping of variables in objects. Model-IT is applicable in the same class of problems as the above mentioned tools and furthermore user can model systems without specific knowledge of the mathematical relations between their parts.

In Zaraza and Fisher (1996; 1997), introduction to proper systems dynamics is mentioned for subjects such as Physics, Biology and Ecology. More specifically Motion models are proposed for Physics, the microbe's reproduction mechanics for Biology and population dynamics study for Ecology. Other subjects that could be studied with Model-IT include Chemical reaction equilibrium, in mathematics didactics of concepts like "rate of change" or "limit".

Model-IT models without rate relationships are simple mathematical non time-dynamic models of the kind that IQON supports (Bliss, 1994b). This class of systems, which are even simpler and not impressive like time-dynamic models, is not unimportant. It includes for example all the four operations word problems that are used in mathematics in the first school years, and in general modelling of chains of relationships. These models could be useful for the introduction of middle and high-school learners to modelling.

MODEL-IT representation conceptual framework

The Model-IT user must think using mainly the terms: "object", "variable" and "relationship". There is evidence that people find it easy to represent systems as interacting objects:

> *Object-oriented programming languages are particularly appropriate for the design and implementation of computer-based models, since they offer a natural mapping to the phenomena being modeled [Kreutzer, 1986]…. Providing students with an object-oriented framework for modeling allows them to think about the phenomena that they are modeling in a more natural way, by matching the interacting objects that they can see in the world with what they see in the modeling environment, instead of having to translate those objects into abstract representations.*
>
> [Jackson, Stratford, Krajcik and Soloway, 1995]

The adoption of a simple object-oriented language as the system's structural representation supports the conceptual modelling efforts of users. The success of object-oriented paradigm in programming supports Model-IT design decision.

In the conceptual framework of functional representation the user has to select from a set of roughly defined quantitative relations. This simplifies the model syntax and even when a wrong selection is made this is possible to be corrected after interaction with the system. Users do not spend cognitive power to formulate the correct mathematical relations or algorithms and save power to concentrate on the modelled physical system. An other advantage of the semi quantitative modelling is the smooth learning curve of the system. Users can explore models in a small time interval without the requirement of programming or advanced mathematics skills.

However, it is possible that the inaccurate execution algorithm of models may cause some confusions, but this statement must be clarified using specific examples.

4.3.3.4 Pedagogical use and assumptions

Dynamic modelling, for many pre-college science students, is an out-of-reach cognitive activity. Model-It provides intentionally designed scaffolding for learners, enabling them to build and test dynamic models of complex systems easily, using object-oriented and qualitative techniques.

The scaffolding strategies of Model-IT intends to:

1. ground the learner in prior knowledge and experience;
2. bridge the learner from novice to expert understandings and practices;
3. couple the learner's mental model with testing actions and model feedback.

Strategy	**Model-It Implementation**
Grounding in Experience and Prior Knowledge	⇨ Pre-defined high-level objects. ⇨ Digitized, personalized photographs and graphics. ⇨ Qualitative, verbal representation of relationships.
Bridging Representations	⇨ Textual to graphical representations of relationships. ⇨ Qualitative to quantitative definition of relationships. ⇨ Concrete to abstract representations of the model.
Coupling Actions, Effects, and Understanding	⇨ Direct manipulation of variable values while a simulation is running. ⇨ Immediate, visual feedback of the effect of user's changes in the values of variable

Table 4.2: Scaffolding strategies and their implementation in Model-It

4.3.3.5 Research results

The use of Model-It in secondary school science classrooms has been the focus of much research at the University of Michigan (Jackson, *et al.*, 1996a; 1996b).

Overall, this research, which has taken the form of case studies, has shown that students use several higher order cognitive tasks when creating models with Model-It, including identifying causal relationships and elaborating upon explanations (Stratford, 1996). Students learn the scientific content that forms the basis of their models as well as ideas related to the nature of science, such as the purposes of modelling (Spitulnik, 1998).

More specifically, Spitulink, Krajcik and Soloway (1999) accounting for one of these case studies which examined pupils' understandings of the phenomenon of ozone depletion, as they engaged in the learning environment of Model-It, state:

"The findings from this study suggest that given a collaborative, inquiry-based model-building environment, students construct scientific understanding. Model building provides a context for students to build and integrate content, inquiry, and epistemological understandings." (p92)

Moreover, concerning the value of Model-It itself they say

"[...] model building would not be possible without the technological support of tools like Model-It. Further design, testing and implementation of these types of tools will help identify the key features that support students in building models, in doing science [...], and in reflecting on the process." (p93)

4.3.3.6 Relations to other modelling tools

a. STELLA

The Model-IT models are mapped in a system of differential equations in order for the model to be simulated. The same structure of the underline formal representation system of STELLA and Model-IT relates them. Both tools are applicable in many same problem domains. The main difference between the tools is that for the simulation of STELLA models the mathematical relations of different variables need to be precisely specified. In the other hand Model-IT can simulate roughly specified relations and give immediate feedback to the modeller during the modelling process. A model is executable even when it is not complete and even when there are not specific mathematic relations between its variables. This characteristic enables modelling of very complex and abstract systems (e.g. social systems).

b. IQON

The relation of Model-IT to IQON (Bliss, 1994b) relies on the semi quantitative kind of models they support. In both tools users are not required to specify detailed mathematical relations for variables but qualitative high-level descriptions.

4.3.3.7 Software engineering issues

MODEL-IT is a standalone PC and Macintosh application with a quite user friendly and ergonomic user interface.

4.4 Analysis of modelling environments allowing creation of qualitative models

4.4.1 Concept maps

Eighteen different software products (either freeware or commercial) were found to have web sites on the Internet. Their contents were analysed and whenever possible, the products were tested. Table 4.2 presents some background information on the different concept modelling software.

Following is a discussion of each of the tested/reviewed software environments.

4.4.1.1 Axon Idea Processor

(http://web.singnet.com.sg/~axon2000/index.htm)

Axon Idea Processor is a Windows program, Prolog based, for visualising and organising ideas, concept mapping, and creative writing. Axon combines Prolog and Object-oriented technology. Its user interface was designed to facilitate the process of idea evolution. Idea processing is concerned with problems and solutions, questions and answers, opinions and facts. *Axon* is a tool for drawing mind maps, planning and presenting. Other features of the package are text processing capabilities, checklists and problem solving hints. *Axon* screen objects can be linked to sound, video, and image files. There is a text box mechanism for explaining in detail the screen objects. These details can be exported and saved to a separate file. Axon supports multiple transparent layers, which become smaller as the user gets further back. This feature allows for three dimensional representations of concepts. Other possible applications are building working models for numerical calculation, for logic diagrams and decision trees.

4.4.1.2 Activity Map

(http://www.timesystem.com/timesystem/software/AMW/Default.htm)

Activity Map is designed and distributed by the Time/System International a/s. By employing a series of memory aids, graphics, colours and symbols *Activity Map* enables users to trace the development of their own ideas and break them down into plans, deadlines and priorities. With a series of memory aids, graphics, colours and symbols the software enables visualisation of the links between the ideas. The structure of *Activity Map* gives users a constant overview of their different thoughts and ideas.

4.4.1.3 Belvedere

(http://lilt.ics.hawaii.edu/belvedere/ and http://advlearn.lrdc.pitt.edu/belvedere/)

"Belvedere" is Java based software (made by University of Pittsburgh) for constructing and reflecting on diagrams of one's ideas, such as evidence maps and concept maps. Belvedere is designed to help support problem-based collaborative learning scenarios in which middle-school and high-school students learn critical inquiry skills that they can apply in everyday life as well as in science. Belvedere constitutes a representational tool for the acquisition of collaborative dexterities in the investigation of real scientific problems. It belongs to the category of those learning environments that mediate collaborative learning interaction and communication of externalized knowledge through an appropriate tool.

4.4.1.4 CLASS

CLASS (Courseware for Learning and Study Skills) is a project at Heriot-Watt University, which has developed a tool to support concept mapping and knowledge reflection.

4.4.1.5 CMap (IHMC CmapTools)

(http://www.coginst.uwf.edu/)

The IHMC Concept Mapping Software (IHMC CmapTools) allows users to construct, navigate, share, and criticize knowledge models represented as Concept Maps. The toolkit is designed and distributed by the Institute for Human and Machine Cognition. The CMap is platform independent and network enabled, allowing users to build, and collaborate during the construction of concept maps with colleagues anywhere on the network, as well as, share and navigate through others' models distributed on servers throughout Internet.

Through a open architecture, the CMap allows the user to install only the functionality required, adding more modules as needed, or as new modules with additional functionality are developed.

4.4.1.6 Decision Explorer

(http://www.banxia.com/demain.html)

Decision Explorer is a tool for managing 'soft' issues such as the qualitative information that surrounds complex or uncertain situations. It allows detailed capturing of thoughts and ideas, provides facilities for their exploring, aiming to gain new understanding and insight. Decision Explorer has been developed by academics at the University of Bath and Strathclyde and now is a trademark of Banxia Software. It provides advanced cause-effect map analysis, enabling the users to enrich the model with new links to concepts. 'Sets' are another important facility used to organise groups of concepts for analysis. Memo cards allow adding background information, perhaps indicating the source of an idea.

4.4.1.7 Inspiration

(http://www.inspiration.com/)

Inspiration is designed and developed by Inspiration Software Inc*.* *Inspiration* is a visual thinking tool helpful for clarification and organisation of ideas and information. It can be used for concept mapping, diagramming, brainstorming, outlining, organising, planning and webbing. There are two editions, the Education and the Professional Edition.

Inspiration *Education Edition* is a visual learning tool that helps pupils to develop ideas and organise their thinking. It uses two main views, *diagram* and *integrated outline* view, that transform pupils' free thinking into a hierarchical structure. By using its *diagram* view pupils are enabled to dynamically create and modify concept maps, webs and other graphical organisers. Switching to *integrated outline* view enables them quickly to prioritise and rearrange ideas, helping them create clear, concise writing. The symbols, templates and example files have been developed specifically for use in the classroom. Visual toolbars make *Inspiration* easy to learn and use because all its major functions are readily available. That is, *Outline* button instantly converts a free-form diagram to a hierarchical outline. *Symbol* and *Create* tools quickly add new symbols to a diagram. Colour buttons customise the fill, line and text colours. *New look* button changes the default symbol and link shapes as well as styling.

Additional features enable pupils to create, rearrange and correct an outline. Dragging topics forces subtopics to move automatically. Discontinuous topic selections allow dragging and working on multiple topics at once. The *Split topic* feature quickly reorganises a topic into two topics and *Promote*, *Demote*, *Collect* and *Move* features help pupils quickly reorganise complex outlines.

Many examples of educational uses -for different grade levels, even for grade 1- can be found at: (http://www.inspiration.com/book/lit_elem.html) Understanding Literary Elements (excerpt from Classroom Ideas Using Inspiration)

Inspiration Professional Edition is a visual learning tool used for knowledge mapping, process mapping, multimedia flowcharting and everyday planning. It is used in many different areas of business, from front-end planning of special projects, Web sites, or proposals to describing processes flow.

4.4.1.8 Kidspiration

(http://www.inspiration.com/)

New to Inspiration Software®, Kidspiration™ helps students from K-5 see, organize and develop their great ideas. Using the proven principles of visual learning, young readers and writers build confidence in organizing information, understanding concepts and expressing their thoughts.

K-5 students use Kidspiration to:

- Brainstorm ideas with pictures and words
- Organize and categorize information visually
- Create stories and descriptions using engaging visual tools
- Explore new ideas with thought webs and visual mapping.

4.4.1.9 Kmap

(http://www.w3.org/Conferences/WWW4/Papers/134)

Kmap is a concept mapping and diagramming tool for Apple Macintosh platform. It has an open architecture that allows different forms of concept mapping to be defined and supports integration with other applications. *Kmap* may be used in a variety of situations, ranging from concept mapping in education through multimedia indexing and semantic nets for knowledge-based systems, to workflow support for scientific communities on the Internet. It is also programmable so that user actions can communicate with other applications. It can facilitate collaborative groupware systems operating on local or wide networks.

4.4.1.10 Mind Manager

(http://www.mindjet.com/)

*Mind Manager (*also called Mind man*)* is a visualisation and organisational tool that offers a variety of ways to produce emphasis in Mind Maps. When a new Mind Map file is created Mind Manager automatically places the main topic box in the centre of the page. Users draw relationships in order to highlight the associations between the ideas. Connection arrows or direct links to other files or sites are allowed.

Mind Manager employs the tree-like organic layout. Several branches and sub-branches of an idea make the total structure. Rearrangement of branches, text or images is allowed with the special drag-and-drop feature. Also its "*Overview*" view allows users to insert their thoughts in a hierarchical structure and then to navigate in this Map on a smaller scale.

4.4.1.11 Mind Mapper

(http://www.mindmapper.com/)

Mind Mapper is a learning software tool that employs a graphics based method of taking notes, brainstorming and organising random thoughts in tree-like diagrams. The subject is represented by a central image. The main themes of the subject radiate from the central image as main branches. Minor themes are linked to the main themes. All the branches are connected forming a nodal structure. It can be used for note taking, creative writing and report writing situations.

4.4.1.12 Mind Mapper Junior

(http://www.mindmapper.com/MMJr.htm)

Mind Mapper Junior is a version of Mind Mapper with a user interface designed for children.

4.4.1.13 MOT

(http://www.licef.teluq.uquebec.ca/francais/)

The software MOT (Modélisation par Objets Typés) is a tool to construct graphical representations.

4.4.1.14 PIViT: Project Integration and Visualisation Tool

(http://www.umich.edu/~pbsgroup/PIViT.html).

PIViT is a product of the Project-Based Science group, University of Michigan The main idea is to help teachers to visualise and plan complex, integrated curricula such as those associated with Project-Based Science. *PIViT* provides graphical mapping tools that support teachers as they move from brainstorming to feasible, integrated project designs. There is a "Project Design window", a space where all the instructional components are drawn and the relationships shown. This window supports sub-mapping, highlighting and colour coding.

4.4.1.15 Representation Tool

(http://hermes.iacm.forth.gr).

Representation Tool is an open ended educational software of collaborative concept mapping, based on the Internet. The essential aim of this software is to provide computational support for the emergence of children's cognitive representations. It comprises of specific components for supporting and recording the representation procedure. Additionally it contains a component for collaborative and distant expression of representations with reactive agents. Furthermore, this software can be used in multiple applications based on the computational concept mapping.

4.4.1.16 SemNet

(http://trumpet.sdsu.edu/semnet.html)

SEMNET is implemented on the Macintosh. It was designed by the SemNet Research Group (Fisher, 1990) and was first introduced into college biology classrooms in 1987.

SemNet is a general-purpose tool with which a user can construct a knowledge representation in the form of a semantic network, that is a network of concepts linked by named relations, with associated text and images. *SemNet*, as a content-free tool, can be used to represent denotative factual or other descriptive information about virtually any domain of knowledge (Fisher, 1992).

4.4.1.17 SMART Ideas

(http://www.smarttech.com/products/smartideas/index.asp)

SMART Ideas is a concept mapping software for Windows developed by SMART Technologies. It aims at helping users to record, organise and communicate ideas by giving them the tools needed to create concept maps. With this software one can: Create concept maps to organize ideas, Expand maps into multilevels, Add links to files and Web sites, Switch between diagram and outline view, Publish your diagrams on the Web.

4.4.1.18 VisiMap Lite

(http://www.coco.co.uk and http://www.dynamicthinking.com/visimap.htm)

VisiMap Lite is designed, developed and marketed by Coco Systems Ltd which develops and supports visual thinking software. It is a tool for visual organisation, brainstorming, problem-solving, document outlining, planning, personal organisation and many other day-to-day tasks. *VisiMap Lite* automatically captures ideas around the central theme, problem, plan, or idea, splitting it into manageable pieces in a tree-like, graphical, hierarchical structure (http://www.visimap.com/ vlvm~compare.html). Users can attach short annotations to branches of a map and as they re-order a map, these notes move with their branches. The user can split maps into modular, linked maps to form multi-dimensional structures.

4.4.2 Analysis of WORLDMAKER modelling tool

4.4.2.1 Historical data

The first version of WORLDMAKER(HK) software modelling environment designed and developed at the Department of Curriculum Studies of University of Hong Kong by Dr. Nancy Law and Mr. Eric W. C. Tam was based on the conceptual design of an earlier work of the same title (WORLDMAKER) developed on a different hardware and software platform in the U.K. by Richard Boohan, Jon Ogborn and Simon Wright, at the University of London, Institute of Education [Boohan, 1992]. The new project was started in 1995 by Dr. Nancy Law in Hong Kong and developed as a collaborative work between the two institutions. The new version of the software was first presented at the International Conference on Computers in Education in 1998 [Law and Tam, 1998].

WORLDMAKER(HK) has been modified in version WORLDMAKER 2.0 in order to exploit computer network infrastructures currently available, which support distributed collaborative modelling activities.

Documentation, research papers, curriculum material, and full functionality trial versions are available at WORLDMAKER(HK)'s web site [WORLDMAKER(HK) web site].

4.4.2.2 WORLDMAKER overview

In WORLDMAKER the user defines objects and their iconic representations (small tiles). User-defined objects are placed on a grid which is called the "world grid". The objects on the world grid are interacting with their neighbour objects according to user-defined rules (Figure 4.19). The rules are not scripted in any language but are specified visually using the "rule editor". Each rule is associated to a firing probability factor, which determines if an activated rule will eventually be applied to produce the next grid state.

Placing very simple objects that interact locally (with their neighbours) in a grid is a simple idea that can express very complex systems such as gas spreading, ecosystems etc.

The simulation of a WORLDMAKER model is a process that produces a sequence of grids from an initial one, by applying the grid object-rules for each state to produce the next. In each state new objects can be placed by the user on the grid, or can be generated by a rule application, and other objects can be removed by the user or automatically.

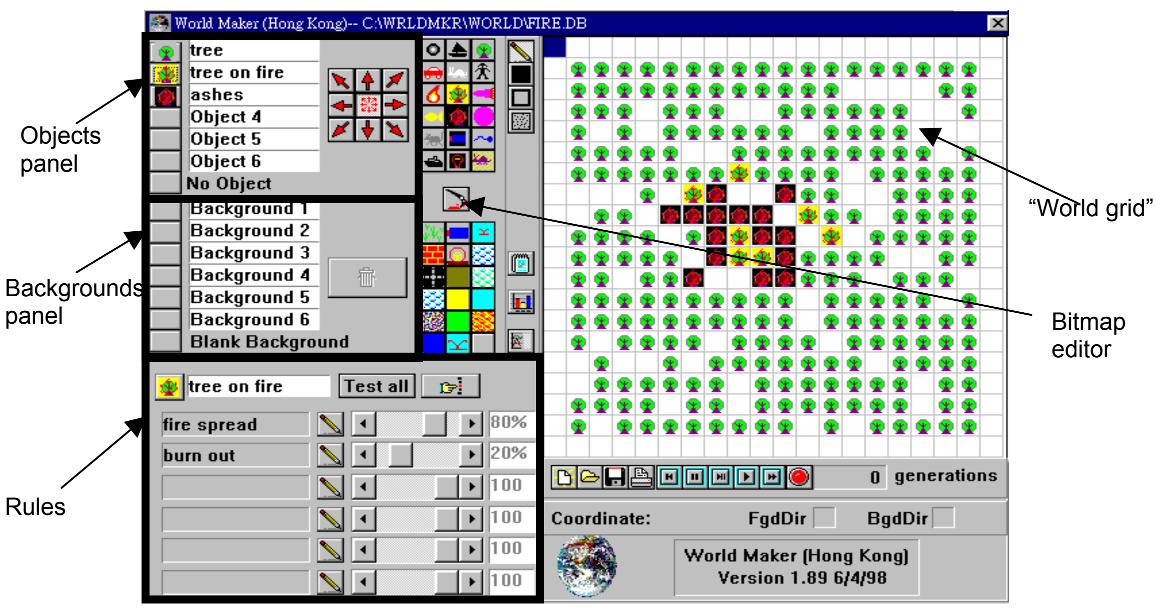


Figure4.19: The WorldMaker interface with a forest fire spread model.

Systems which have a large number of objects interacting in parallel are very complex and diverse and thus difficult to model with other formulations even with programming languages because of their sequential nature.

4.4.2.3 Underlying representation system

WORLDMAKER’s representation system is based on the cell automata computation formalism [Toffoli and Margolis, 1987] and can be used for the study of systems with macroscopic deterministic behaviour, produced by a large number of microscopic and even stochastic parallel interactions of the system particles. Many people know about cellular automata because of Conway’s [Gardner, 1970] popular mathematical “Game of Life”. Essentially a cellular automaton consists of an array of cells, in which each cell can exist in one of a number of possible states. The cellular automaton evolves over time, and for every iteration *rules* are applied to each cell in turn to determine its state in the next generation. The rules which determine the subsequent state of a cell are based on the states of the adjoining cells.

Cellular automata, genetic algorithms, artificial life, classifier systems and neural networks are computing paradigms that offer new modelling opportunities and thus have contributed significant developments in biology and other sciences. All these representations deal effectively with parallel computation formulation. The study of these representations is known with the term parallel intermaths [Bailey, 1996].

a. What systems’ domains does the tool properly affront?

Cellular automata constitute a modelling formalism proper for systems of parallelly and locally interacting cells. The systems properly represented by cellular automata show extremely complex overall behaviour - yet are constituted by very simple and similar component parts. The complexity source is the large number of parallel interactions between the system’s components [Wolfram, 1984].

Examples of physical systems of this kind are:

1. Traffic
2. Chemical reactions
3. Gas spreading
4. Virus Spreading
5. Ecosystems
6. Crystallizations (annealing)
7. Electrical current
8. Flock of birds, fishes etc
9. Earthquakes
10. Economics
11. Dynamic equilibrium systems

These systems could be studied macroscopically using other representation systems like differential calculus, but this approach is not concerned with the mechanisms which create the overall behaviour of the systems. The macroscopic study of these systems is not enough for the understanding of the intermath systems. Furthermore the cellular automata representation is much easier to learn and use than the differential calculus one, thus making the study of these complex systems possible at the secondary education level.

b. WORLDMAKER representations’ conceptual framework

The WORLDMAKER models are objects that the user defines and places on a grid (the world grid) to interact. *“The reason for introducing objects is that a cellular automaton* *does not make it easy to model an entity which has a continued existence and which can move around, so objects provide this role."* [Boohan, 1992].

There are two kinds of objects: foreground and background - the only difference between them is that background objects cannot move on the grid. The interaction is specified by user-defined rules that are visually constructed. In order to specify a rule the user must specify what the object's neighbourhood must be like for the rule to activate and what is the new state of the object after the rule fires.

In general, many of the models which can be created with objects are probabilistic and draw on, for example, the statistical bead games described by Eigen and Winkler [1983]. Marx [1981a, 1981b, 1984a, 1984b] has produced many examples of these kinds of model for microcomputers, but the rules are part of the program and are not accessible to the user. WORLDMAKER allows the user to modify and create the rules governing behaviour. Another interesting program, 'The Picture Simulator', exists [Camara *et al.*, 1991], which also allows rules to be defined for entities, but it takes a rather different approach. It uses a more restricted range of rules and kinds of entities than WORLDMAKER, and its interface has not been designed to make the program accessible to young children.

Moreover, for a quantitative study of a system's behaviour WORLDMAKER has a statistics module that can be used for the production of diagrammatic representations of several quantities.

The conceptual framework is quite simple and powerful. There is good evidence that young students can internalize this object-event framework and can use it to express their ideas [Boohan, 1994].

c. Modelling with WORLDMAKER

There are essentially three levels of modelling at which WORLDMAKER may be used for [Boohan, 1992; Boohan, 1995]:

'Exploring a world': Making models from foreground and background objects with predefined behaviours.

'Changing a world': Modifying the behaviour of objects by changing their rules.

'Creating a world': Creating entirely new types of objects by building a new set of rules.

4.4.2.4 Epistemological theses

For WORLDMAKER(HK) designers: *"A model is a simplified and idealized system that can be used to think about another. Key concepts and principles in many disciplines are often theoretical models of phenomena under scrutiny"* [Law and Tam, 1998].

Theorizing or models' building is nowadays considered an important part of scientific work. However, understanding the role of modelling in science is not easy for learners, unless they explore the consequences of different theoretical models. Information technology provides environments which permit the efficient exploration of theoretical models with computerized modelling tools.

Computerized modelling paradigms and tools affect model building significantly. Good modelling tools provide structures that help express thought, so new thoughts about the world could emerge from the interaction with the model (artificial external world).

Thus, WOLRDMAKER is a modelling tool based on a new powerful modelling paradigm (cellular automata) of non-trivial practical use, allowing the exploration of related domains of problems.

4.4.2.5 Pedagogical theses

Beyond the specific to the didactics of physics assumptions embedded in WORLDMAKER's approach, there are also some general pedagogical theses concerning the process of modelling such as:

"Modelling as a learning activity in science would help students to develop an understanding of scientific principles as theoretical models and not facts. It would also help develop a better understanding of nature of science and a deeper understanding of the specific principles explored." [Law, 1999, p1]

WORLDMAKER's pedagogical approach adopts Einstein and Infeld's view that *"Science is not a collection of laws or a catalogue of facts but a creation of human mind with freely invented ideas and concepts"* [Einstein and Infeld, 1938] and assumes that for learners to develop such an understanding they should have the opportunity to explore the consequences of different theoretical models [Law and Tam, 1998, p2].

About the educational use of computer-based modelling the WORLDMAKER's approach assumes that:

"Computer-based modelling… allows the user to externalize thoughts as well to interact with them: computer-based models are "runable" thoughts" [Law and Tam, 1998, p2].

Concerning the pedagogical value of computer modelling tools the new developers say:

"Modelling activities… provide opportunities for children to engage in activities involving high level of cognitive understanding and creativity in making sense of phenomena and possible contributing (whether tangible or abstract) factors for the domain context. The provision of appropriate modelling tools to enable children to create their own representations ("worlds") and to explore other people's representations would greatly enhance the accessibility of such activities to younger children" [Law and Tam, 1998, p2].

The above assumptions could be located in most educational modelling approaches.

A more specific assumption which relates to the representation system embedded in WORLDMAKER is that young children find the creation of models based on objects and events accessible [Boohan, 1994].

4.4.2.6 Curriculum use of WORLDMAKER

The teaching of systems of the kind mentioned above can be found in several areas of the school curricula. Indeed, in conjunction with the first prototype versions of WORLDMAKER a number of 'learning tasks' were developed related to the areas of science, geography and maths. According to their authors [see Boohan, 1992, pp8-9]:

"In science, these kinds of models [i.e. WORLDMAKER models] *are particularly appropriate in those topics where global behaviour emerges from collections of interacting objects, which are often organisms (in biology) or atoms and molecules (in physics and chemistry). Such topics include population studies, behaviour of organisms, diffusion, crystallisation and chemical kinetics and equilibria. In geography, models are often concerned with* flows*, and so involve* directional objects - *examples include coastal erosion, volcanoes, rain and different soil types. Applications in maths are rather more heterogeneous, and include number patterns, algebraic representations of series, graphical representation of changes over time, and probability."*

WORLDMAKER is also recently mentioned in the list of computer-based modelling tools of the module "The rise and fall of the Clockwork Universe" of AS/A2 syllabus specification for Physics ('Advancing Physics') created by the Institute of Physics in the UK. The relationship of WORLDMAKER with the didactics of physics is summarized in the following scripture by Law [1999, p3]:

"Physicists appreciate that many natural phenomena are currently explained by assuming that matter consists of microscopic particles, which act in certain, well-defined

ways. Prime examples of this include diffusion and Brownian motion, both of which are explained by the kinetic theory.

In a school laboratory situation, it is often difficult to explain how these abstract principles in Physics relate to the macroscopic phenomena observed in standard experiments. There are some physical models that attempt to provide analogies for illustrating these microscopic interactions. However, such models tend not to be very effective as the physical properties of the "particles" used in these models, like beads and beans, are rather different from the "ideal particles" that they simulate.

Given that the structure of matter is one of the most fundamental and mind-boggling theories about the universe (the atomic theory started with the Greeks, yet did not convince many scientists until the early 19th century, and was not proven until 1905 by Einstein), it is important and yet not easy for the students of today to fully comprehend the nature and implications of such a theory. By using a cellular automaton simulation system, one can make it easier for students to realize how the macroscopic phenomena may result from the random interactions of microscopic entities. While physical models of the kind described above are effectively only illustration tools, students can actually explore the various features of the system using WM. This allows them to make sense of and construct their own understanding of how a theoretical model of microscopic entities can in fact produce deterministic macroscopic behaviour, fostering the development of deep cognitive understanding and meaningful learning."

However, as its authors themselves mention, WORLDMAKER should not be seen as just a tool to help support specific topics, since it is also intended that children will come to a better understanding of computational modelling.

"We have used the metaphor of a 'world' to mean a model, since it invites comparisons with the real world. So, to create their own world, children need to consider what elements of the real world are relevant. They need to make their own world simpler than the real *world. Has making the world simpler left out important elements? Or are these just a distraction from the essential features? They need to consider to what extent their own world behaves like the real world. Why does the computer world sometimes behave unexpectedly?"*

4.4.2.7 Limitations of WORLDMAKER

WORLDMAKER is best used for studying populations of objects (e.g. collections of people, animals, plants, molecules, etc.), each individual instance of which obeys the same set of fixed rules. It completely excludes "action at a distance" and thus cannot model objects that interact through force fields, for example. Moreover, Newtonian motion cannot be simulated with any exactness, although some features of it, such as collision can (Ogborn, 1999).

4.4.2.8 Research results

The original development team tried out WORLDMAKER mainly with younger pupils from 9 years up to 12 years (Boohan, 1994; Boohan and Maragoudaki, 1997; Maragoudaki, Boohan and Ogborn 1997; Maragoudaki and Boohan, 1999; Ogborn, 1999). The results of the trials show that pupils of this age group begin to learn to use the software quite quickly, but take longer to master all it can do.

More importantly however, they found that *"the idea of making a model by giving objects rules turns out to be easy to understand, and students rapidly learn to make an interpret rules in pictorial form"* [Ogborn, 1999, p23].

Research results from the new development team can be found on the WORLDMAKER web site [WORLDMAKER web site] and refer to older pupils. Most of them are from observation of students groups working with the tool in several modelling activities carried out in the framework of Shum's (unpublished) MEd dissertation at the University of Hong Kong [Shum, 1997]. The main goal of the research was the illustration of the kind of thinking that has been stimulated by modelling activities using WORLDMAKER. The results are expressed in the following extracts:

"It has been generally found that discrepancies between prediction and observation provided very stimulating contexts for discussion and further exploration. Comparison of situations where students were using the software in individual settings with those in group settings consistently revealed that the latter led to more fruitful learning. Group discussions provide a non-threatening environment for expressing and exploring ideas, allowing deeper understanding to develop." [Law and Tam, 1998, p7]

In the same paper it is mentioned that WORLDMAKER can be used also as a cognitive scaffold to support problem solving.

Shum's initial WORLDMAKER trials show that it helps students to externalize their thinking and resolve some of their misconceptions. Previous trials also show evidence that the software may encourage abstract thinking even in younger pupils, who with the help of the pictorial expression of rules can identify patterns of similarity between different models. (Ogborn, 1999, p24).

A common difficulty to younger and older pupils was deciding which elementary actions available in the system to use to make a model. *"Where an effect [...] has to be made out of several rules, the way forward is often not clear. Simple, direct rules that correspond well to the way the actual situation is imagined [...] are much easier to arrive at."* (Ogborn, 1999, p23)

Finally, Law and Tang (2000, pp2-3) mention that teachers that used the software found it difficult to motivate students to build their own models. Subsequently, teachers proposed that models should be put in a goal-directed games form to engage students. This criticism has been a development goal for WORLDMAKER 2.0.

4.4.2.9 Relations to other modelling tools

a. AgentSheets

AgentSheets system is a computerized modelling environment maintained by AgentSheets Inc. AgentSheets is the brainchild of Dr. Alexander Repening and was born in the Center for LifeLong Learning and Design of the University of Colorado. [Repenning, Ioannidou, and Zola, 2000; AGENTSHEETS web site].

As a commercial product AgentSheets has the form of a mature software implementation. The connection of AgentSheets with WORLDMAKER is that they are based on the same representation system, the cellular automata. AgentSheets provides the capability of distributed collaborative modelling in terms of different group-constructed objects that can be put to interact in the same shared world. Imagine for example that two groups of modellers at different locations have designed different car-objects that were then put to race in the same world. This kind of collaboration permits even competition and would be very interesting to explore from a learning point of view.

b. StarLogo

StarLogo is developed in Media Laboratory of MIT, with the support of the National Science Foundation and the LEGO group. StarLogo is based mainly on Resnick's ideas about decentralized modelling systems. That is systems that are organized without an organizer or coordinated without a coordinator. Main real-life phenomena like bird flocks, traffic jams, ant colonies, ant market economies are examples of decentralized systems [Resnick, 1991; Resnick, 1992; STARLOGO web site].

It is obvious that the target systems of StarLogo and WORLDMAKER are similar. StarLogo is based on the LOGO language representation system. The concept of 'object' used in WORLDMAKER corresponds to the turtle of StarLogo. Many turtles can be put in a world to interact according to the rules that the modeller will script in LOGO language. It is obvious that an experienced programmer or modeller has much greater programming power with StarLogo than with the simple conceptual framework of WORLDMAKER.

However, the familiarization with StarLogo is probably much more difficult than with WORLDMAKER or AgentSheets for most students. StarLogo is better for the modeller who knows some LOGO programming and needs more power than what the WORLDMAKER can offer. For example WORLDMAKER "world-grid" has a fixed and relatively small size and the number of different objects and rules governing each of them is limited to 6.

4.4.2.10 Software engineering issues

WORLDMAKER(HK) is a standalone windows application with a quite user-friendly and ergonomic user interface. It supports two languages: English and Chinese. WOLRDMAKER has a world grid of 20x20 tiles and permits the definition of at most six kinds of foreground objects and six kinds of background objects with at most six rules governing the behaviour of each object kind (that is 72 rules at most for each model).

The user specifies a rule using the "rule-editing panel". This is a graphical rule editing tool which removes the need for a scripting programme.

WORLDMAKER 2.0 will exploit the communication network infrastructures in order to permit distributed modelling collaboration.

4.5 Implications for the design of a modelling tool

Based on the above review of existing computer-based modelling environments we can draw a number of (general and specific) implications for the design of a new modelling tool. According to these the new software (amongst other things) should allow for and facilitate:

- Modelling of different categories/classes of models, e.g. boolean, algebraic, differential.
- Concept mapping. To create, rearrange and correct a concept map easily and flexibly.
- Separate formalism for representation of semi-quantitative information; this formalism should be regarded as essentially distinct from classical mathematics, i.e. closer to 'commonsense knowledge'.
- Modelling of conditional events, i.e. of the occurrence of some events conditioned by other events.
- Modelling of direct relationships as well as rate of changes.
- Iconic representation and animation of the model, so that the user can visualise the changes to the system over time.
- Real-time interaction with the model, while it is running.
- A variety of output representations: animation, tables, graphs, numeric display of results. Discussing and linking the ideas imbedded in these different outputs can help the process of creating meaning from different representations.
- Visual representation of phenomena with a lesser degree of formalism than equations, tables, or graphs.
- The import and analysis of experimental data within the environment. E.g. the user may import measurements collected experimentally and produce relevant graphs to compare with graph(s) produced by the model.
- Appropriate feedback to users interacting with underlying models.
- Incorporation of scaffolding strategies: ground the learner in prior knowledge and experience; bridge novice to expert understandings and practices; couple learner's mental model to testing actions and feedback.

- Playing back the construction of a model so that people can trace the development of their ideas, and also people not involved in constructing the model can build an understanding of its logic, and associated dynamic behaviour, in steps.
- Different calculations of the combination of inputs, e.g. average or sum of input values.
- The import of still pictures and if possible movies from external (to the programme) digital libraries.
- Loops of links to be disallowed.
- Differentiation between an inverse link (1/x) and a subtractive link (-x).
- Representation of the initial quantities of a variable.
- Visualisation of negative values and variable scales.
- Graphs being constructed in real-time.
- Annotation of the reconstruction of the model by associating other kinds of representations to it including text notes.
- Attachment of short annotations to branches of a concept map, which they would move with their branches if the map is re-ordered.
- Both expressive and exploratory modes of learning.
- Mediation of collaborative learning interactions and communication of externalised knowledge.
- Modelling a wide range of problems related to secondary school curriculum.
- Presentation of information only appropriate for students' knowledge and age-range.
- Pupils to think about the phenomenon in its entirety and create the respective model, without having to be mathematically precise about the relations between, and functional behaviours of the variables involved.

On the whole, it was noted that the current tendency is for development of exploratory environments, which are meant to be used as cognitive tools accompanied by appropriate materials. Semi-quantitative modelling environments seem to have smooth learning curves. Pupils learn how to explore and create models quickly, without the requirement of programming or advanced mathematic skills.

Moreover, an object-oriented framework for modelling seems to allow pupils to think in a kind of 'commonsense' way about the phenomena, by matching the objects and interactions that they can see in the world with what they can model in the modelling environment, rather than have to translate them into abstract representations or variables.

5. Review of Technology-based Collaborative Learning Environments

5.1 Definition of collaboration: Collaboration vs Cooperation

"Collaboration" refers to the fact that a group of people work together on a task. Much has been written about how best to define 'collaborative learning'. A frequent point of departure is to draw a distinction between two terms that are often used interchangeably: 'collaborative' and 'cooperative' learning. The main difference between these terms concerns the nature of the task being carried out and the role of the group members in achieving the task. In a collaborative learning process, two or more people are required to learn something together; what has to be learned can only be accomplished if the group works in collaboration. Communication and negotiation are fundamental in a collaborative learning process. In contrast, cooperative learning requires a division of tasks among group members, where each person is responsible for a portion of the problem solving.

5.2 Theoretical justification for collaboration: from collaboration to knowledge building

Traditionally, cognitive theories have examined learning as an individual and mental process. Scientific thinking has traditionally been seen as a characteristic of an individual mind. However, in explaining human intelligent activity, both cognitive theory and the current philosophy of science increasingly emphasize the socially distributed (or shared) nature of cognition, a process in which individual cognitive resources are shared socially in order to extend them or to accomplish something that an individual agent could not achieve alone. Human cognitive achievements are based on a process in which an agent's cognitive processes and the objects and constraints of the world reciprocally affect each other. Higher cognitive accomplishments presuppose that an agent uses the external world and his or her fellow inquirers as sources of knowledge, organizers of activity, and in general as extensions of his or her cognition.

In the background to theories concerning socially distributed cognition there are observations, according to which many cognitive problems, which cannot be solved individually, can be addressed by combining the limited knowledge and skills of several agents. A fundamental source of advancement of inquiry is social communication and in the context of science scientific argumentation. Through social interaction, the contradictions, inconsistencies and limitations of an agent's explanations become available because it forces the agent to perceive his or her conceptualizations from different points of view. Limited cognitive resources can be overcome by distributing the cognitive load to several agents, each of whom is equipped with a restricted power of cognition. Externalization is an important prerequisite for socially distributed cognitive achievements: as a part of objective knowledge, externalized conceptions can be compared with the conceptions of the others.

A socially distributed process of inquiry provides strong support for the development of the participants' metacognitive skills. So, collaborative learning in which thought processes are externalized in the form of public discourse, provides an agent with access to other participants' processes of thought, thus supporting the development of the agent's metacognitive skills.

5.3 What is a Computer-Supported Collaborative Learning (CSCL) Environment

Computer Supported Collaborative Learning (CSCL) has grown out of wider research into Computer Supported Collaborative Work (CSCW) and collaborative learning. CSCL is used in educational settings and focus on what is being communicated. Computer Supported Collaborative Learning Environments aim at providing both an authentic environment and multiperspectives that can tie in students' prior knowledge. These systems help collaborating parties ascribe mental models to each other, keep such models consistent and support parties to plan actions and act consistently towards a *common shared goal*.

The role of computers in collaborative learning

Computer's use shift learning from recitation to exploration and construction, from being individual-based to being team-based, and from being separated by disciplinary lines to being interdisciplinary. The role best accomplished by the computer and computer networks for collaborative learning is that of affording new opportunities: facilitating students' collaboration even in situations where there are no opportunities for face-to-face communication, shy or less confident children may be benefit from the communication through computers, creation of powerful tools for joint writing and knowledge sharing.

5.4 Collaboration and Requirements from advanced Collaborative Learning Environments: Review of design considerations related to advanced tools supporting collaborative learning

In the classroom, effective collaboration with peers has proven itself a successful and uniquely powerful method (Brown and Palincsar, 1989; Doise, Mugny and Perret-Clermont, 1975). Students working effectively in groups encourage each other to ask questions, explain and justify their opinions, articulate their reasoning, and elaborate and reflect upon their knowledge, thereby motivating and improving learning. These benefits however, are only achieved by active and well-functioning learning teams (Brown and Palincsar, 1989; Jarboe, 1996; Salomon and Globerson, 1989). Katz (1996) argues that despite an abundance of studies showing that social learning situations correlate with a wide range of positive outcomes – including greater learning, increased productivity, more time on task, transfer of knowledge to related tasks, higher motivation and heightened sense of competence (Slavin, 1990) — there is also widely recognized room for improvement.

Placing students in a group and assigning them a task does not guarantee that the students will engage in effective collaborative learning behaviour. While some peer groups seem to interact naturally, others struggle to maintain a balanced participation, leadership, understanding and encouragement. Dysfunctional group activity devalues the benefits of collaborative learning. Collaborative learning does not work for all learners and the result of instructional outcome studies are mixed (Webb, 1987). Fruitful students' interactions are simply not a given. In Brown and Palincsar's (1989) words: *"Social interactions do not always create new learning; peer interactions vary enormously".*

Success in collaboratively learning subject matter means both learning the subject matter (collaborating to learn), and learning how to effectively manage the team interaction (learning to collaborate) (Soller and Lesgold, 2000). Supporting on-line learning teams means supporting both these activities. Educational researchers and technologists developing CSCL tools (Capozzi, Rothstein and Curley, 1996) agree that group members do not necessarily have the social interaction skills they need to collaborate effectively. Students learning via CSCL technology need guidance and support on-line, just as students learning in the classroom need support from their instructor.

In the rest of this section we will describe in brief some existing CSCL systems and comment on the main tools, functionalities that computer supported collaborative learning environments must have for supporting learners that collaborate via CSCL technology.

5.4.1 Brief description of some existing CSCL systems.

5.4.1.1 DIALAB

DIALAB (Moore, 1993) is a synchronous communication tool, designed to teach argument and critical thinking skills using a rigid logic-based dialogue game.

Students use the Dialab system in pairs. It has a set of dialogue moves (allowable move type), a set of commitment rules (used to monitor the statements each player has committed to during the dialogue) and a set of dialogue rules (which determine which move types can follow each other). When using the system one of the students starts by choosing a move from the set of sentence openers. Turn-taking is rigidly enforced and a participant can only make one dialogue move per turn. Following a turn, the computer system updates the sender's commitment store (using the commitment rules) and the move and statement records. The commitment rules keep track of each participant's commitment store and these are updated following each turn and are visible to both participants. Finally win-lose rules are applied to the collaborative dialogue, identifying situations in which a participant ha won or lost the game (for example by showing inconsistency in the commitment store).

5.4.1.2 BELVEDERE

Belvedere's (Suthers and Jones, 1997) core functionality is a shared workspace for constructing "inquiry diagrams" which relate data and hypotheses by evidential relations (consistency and inconsistency).

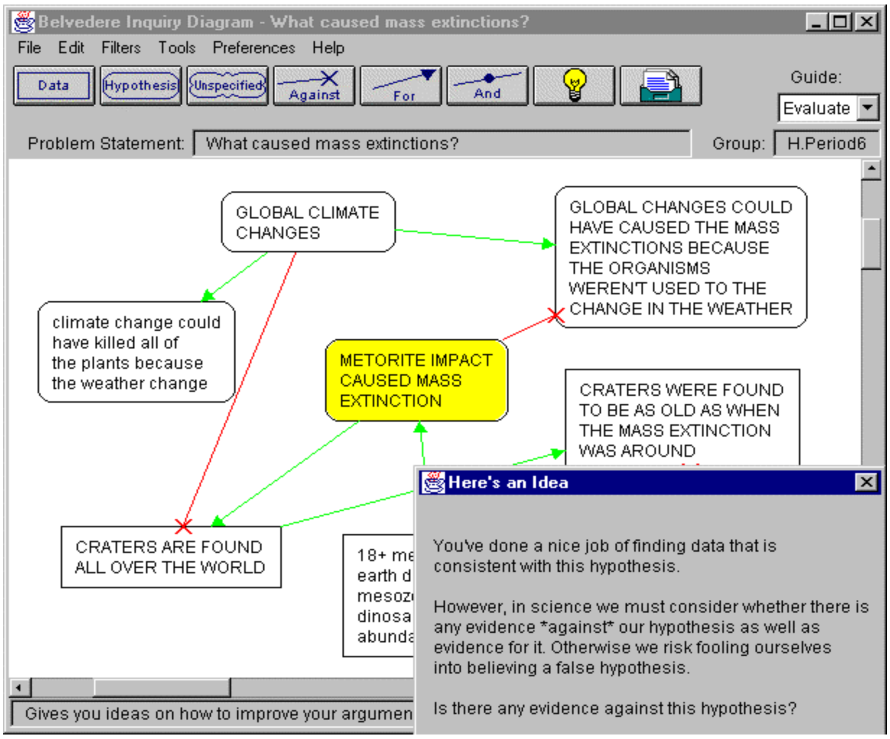


Figure 5.1: Belvedere Inquiry Diagram and Advice

The software also includes a "chat" facility for unstructured discussions, facilities for integrated use with Web browsers, and two artificial intelligence coaches. The first one provides general advice on the structure of the inquiry diagrams from the standpoint of scientific argumentation. It helps the students to understand the principles of inquiry. The other coach performs various comparisons between the students' diagrams and an inquiry diagram provided by a subject matter expert. This coach can provide students with feedback concerning correctness, or confront students with new information (found in the expert's diagram) that challenges students in some way (Figure 5.1).

5.4.1.3 CoVis

CoVis, (Learning through Collaborative Visualization) (Pea *et al.*, 1995), explores issues and scaling, diversity and sustainability as they relate to the use of networking technologies, to enable high school students to work in collaboration with remote students, teachers and scientists (e.g. Figure 5.2 and 5.3).

Participating students study atmospherics and environmental sciences through inquiry-based activities. CoVis provides students with a range of collaboration and communication tools. These include: desktop video teleconferencing; shared software environments for remote, real-time collaboration; access to the resources of the Internet; scientific visualization software; and a multimedia scientist's notebook.

The notebook is a medium for students to record their thoughts and actions as they perform scientific inquiry. A notebook consists of a title page with a brief description of the notebook's purpose, a table of contents and any number of content pages. The table of contents for a notebook displays the notebook's title, a list of its authors and an overview of its pages including their types, titles and relational structure.

Each page in a notebook has a type, a title and a set of authors. Pages may be authored by individuals, or by a group of people working together at the same time. The types given to pages by their authors, provide both description of their contents, and of their relationship to other pages (Figures 5.2 and 5.3).

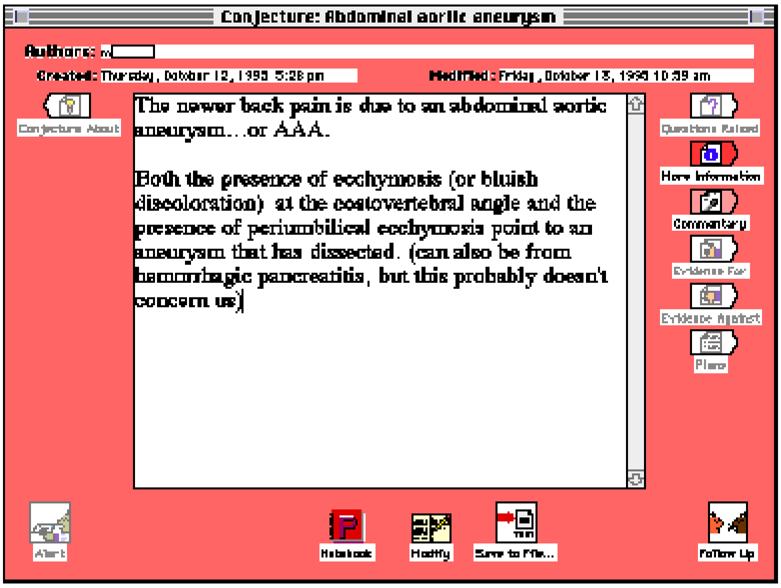


Figure 5.2: A Conjecture page. Links to other pages are represented by the buttons on the left and right side of the text.

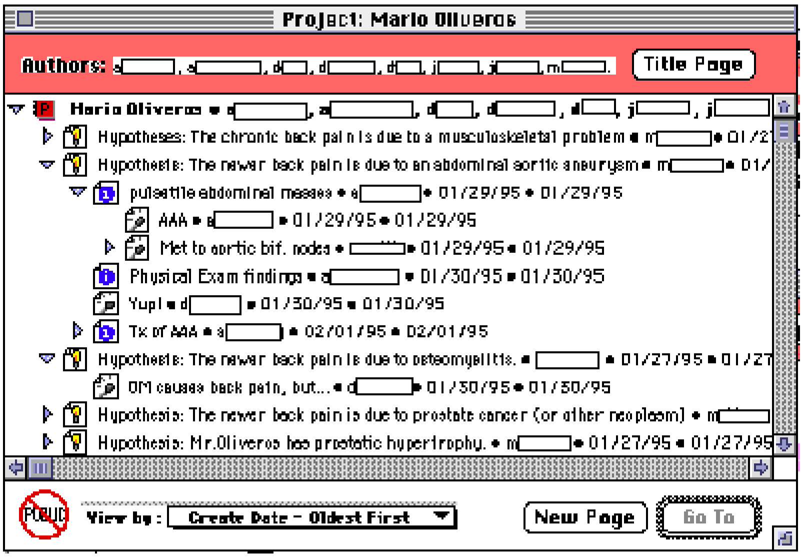


Figure 5.3: A notebook table of contents created in a Northwestern Medical School Problem-Based Learning (PBL) course. Page types are indicated by icons; links are indicated by indentation.

5.4.1.4 COLER

COLER (Suthers, 1998; Suthers *et al.*, 1997) is a web-based collaborative learning environment in which students can solve database-modelling problems while working synchronously in small groups at a distance.

COLER is designed for sessions in which students first solve problems individually (private workspace) and then join into small groups to develop group solutions (public workspace). The initial problem-solving helps ensure individual participation and provides differences between students' solutions that form the basis for discussion. When all the students have indicated readiness to work in the group, the shared workspace is activated, and they begin to pace components of their solutions in the workspace. Only one student can update the shared workspace at a given time. A panel shows the name of the student who has the control of this area and the students waiting for a turn.

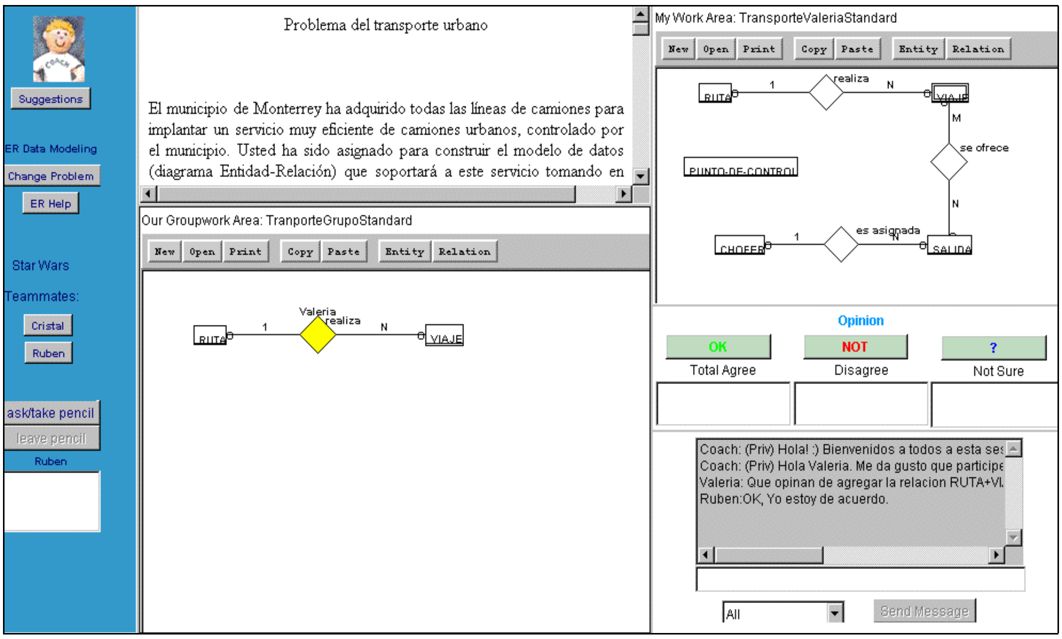


Figure 5.4: COLER Collaborative Student Interface

The students can get information about Entity-Relationship Modeling while COLER's personal coach encourages students to discuss and participate during collaborative problem solving. It observes participation in the shared work-space and in chat discussions. Using this information COLER decides whether to give advice (Figure 5.4).

5.4.1.5 Knowledge Forum

Knowledge Forum (Lehtinen *et al.*, 1998, Scardamalia *et al.*, 1989) allows users to create a knowledge-building community. Each community creates their own database in which they can store notes, connect ideas, and "rise-above" previous thinking.

Users start with an empty database to which they submit ideas, share information, reorganize the knowledge. They can enter text and graphic notes into the database on any topic they are working on. All users on the network can read the notes and users may build on, or comment on, each other ideas. Authors are notified when comments have been made or when changes in the database have occurred. 'Knowledge Forum' makes information accessible with multiple vantage points and multiple entry points. Even the display of the community 's work can be organized in flexible visual displays (Figures 5.5 and 5.6).

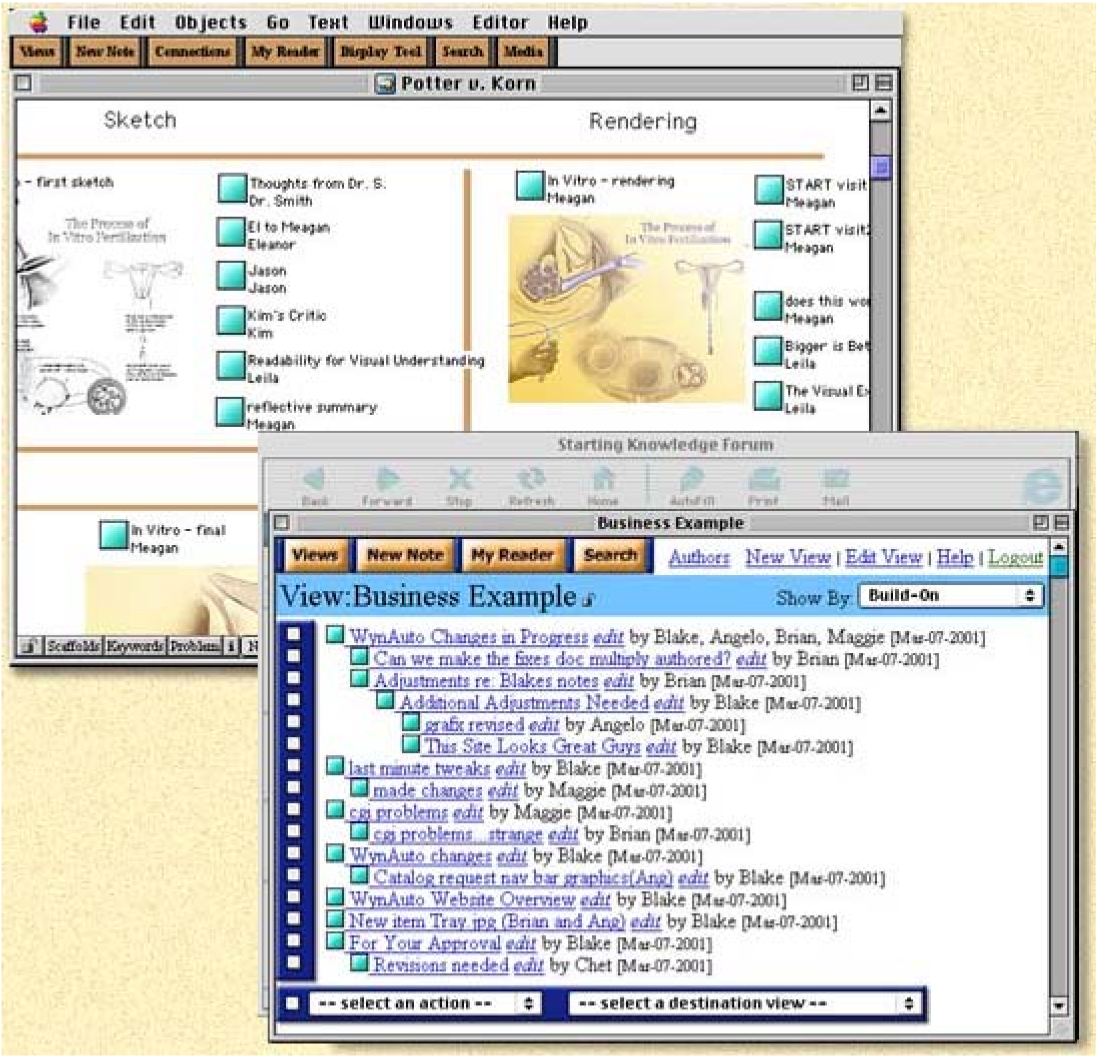


Figure 5.5: Users view the knowledge base from multiple perspectives, thus discovering new connections.

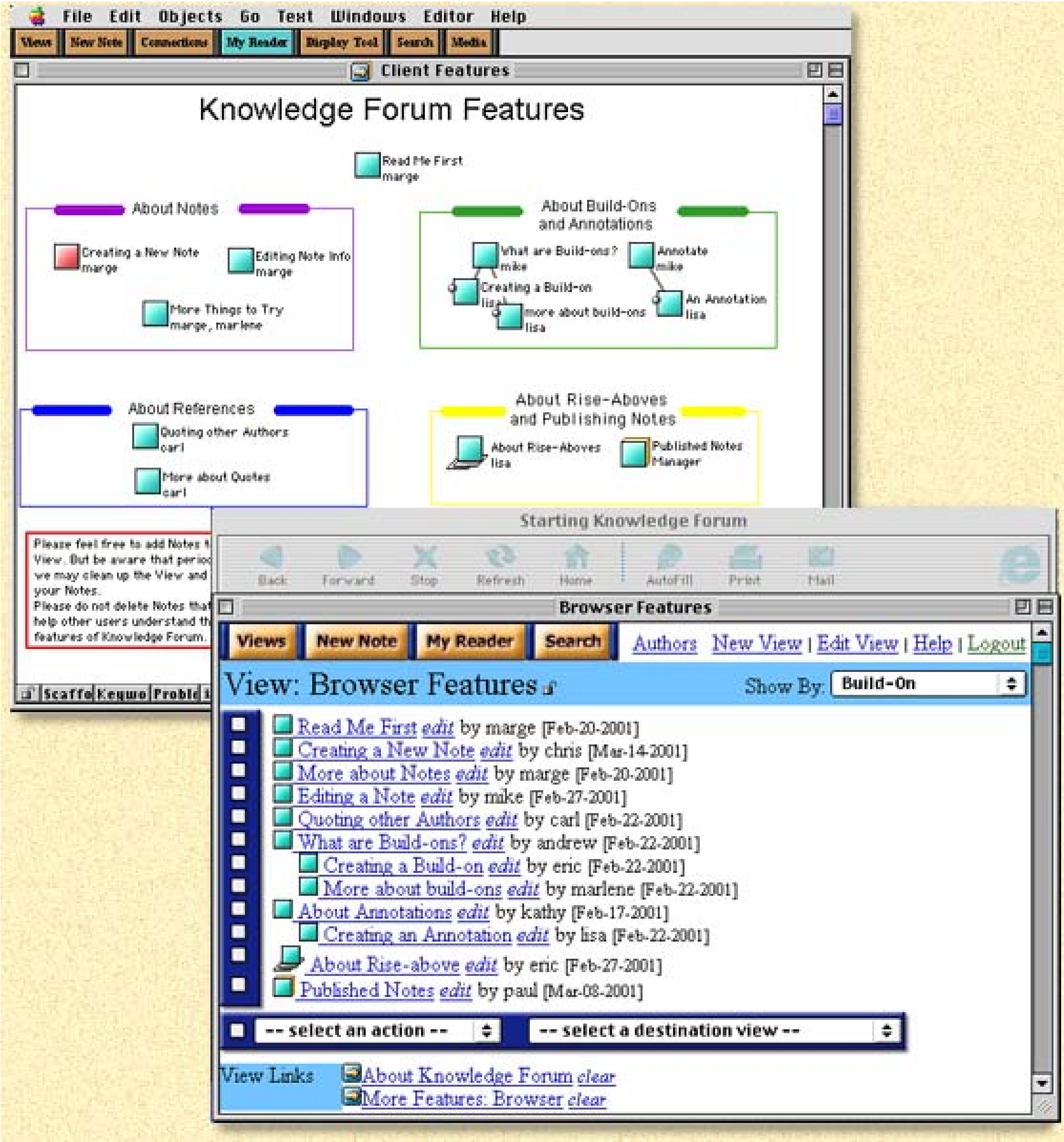


Figure 5.6: Knowledge Forum Features.

5.4.1.6 C-CHENE

C-CHENE (Baker and Lund, 1997) is a CSCL environment for teaching modelling and the concept of energy in physics. The students (16-17 years old) construct their energy chains together in this graphical interface and all of their discussion takes place via specially designed communication interface.

The full screen is divided into two parts from top to bottom, by two buttons for shifting "mode" between "construct" ("construire") energy chains and "communicate" ("communiquer"). In "construct" mode menus appear which contain items for graphically constructing energy chains ("create", "delete", "move", etc.), and use of the lower "communicate" area is blocked. The "communicate" area is activated by the button "communiquer", which blocks construction above by hiding the menus. The "communicate" screen area contains three windows: one chat-box for each of the students and a dynamically updated interaction-history trace. Students type their messages in their respective chat-boxes, then 'send' them by hitting 'Tab key', which clears the message in their box, adding it to the end of the interaction history. It also closes their own chat-box and opens that of the other student. The student can observe all actions on screen of each other, in real time.

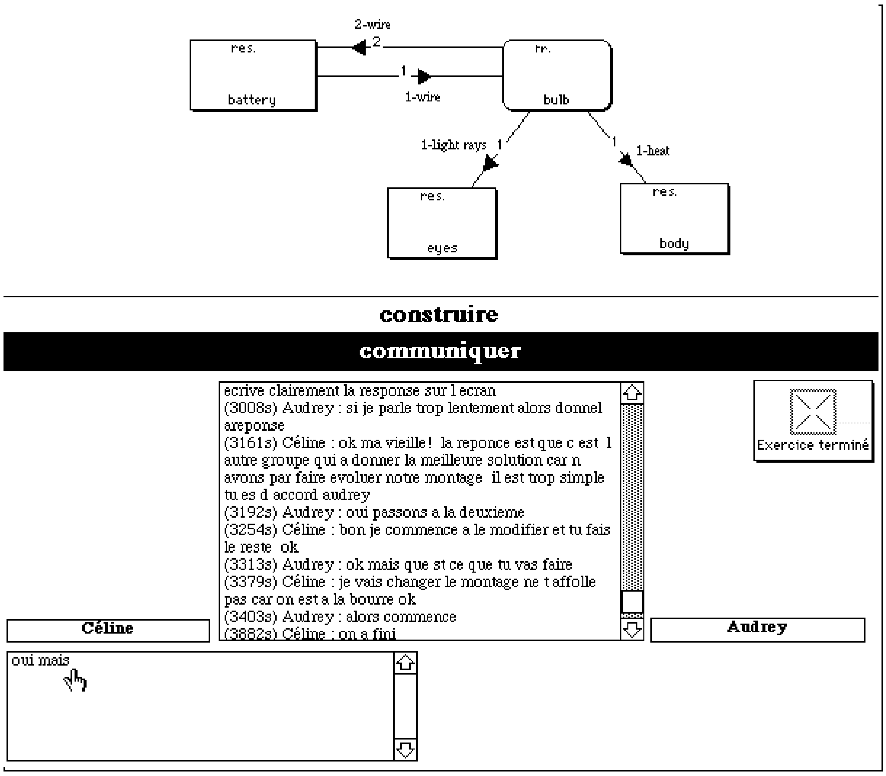


Figure 5.7: The "chat-box" interface of C-CHENE, with students' solution for battery-wires-bulb experiment. The energy chain has been relabelled in English, the text left in the original French.

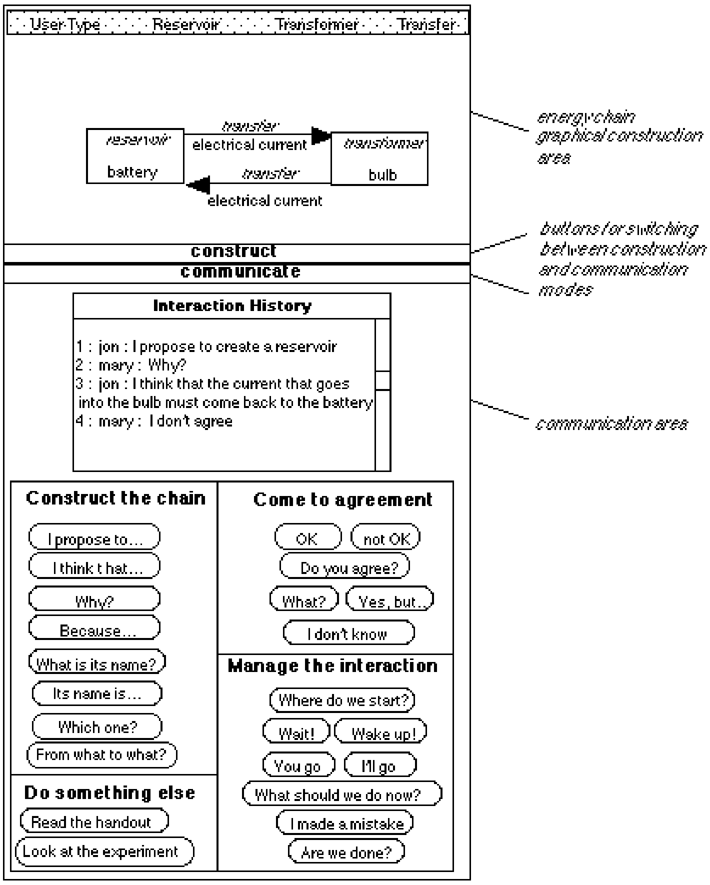


Figure 5.8: C-CHENE "dedicated" communication interface

The interface enforced turn-taking. The possibility of interruption is provided in all situations, and is performed by clicking on the "communiquer" or "construire" buttons in the middle of the screen (Figures 5.7 and 5.8).

5.4.1.7 DEGREE

Degree (Barros and Verdejo, 2000) is an asynchronous CSCL system which characterizes group and individual behaviour in terms of a set of attributes. The system analyzes the state of collaboration using a model of interactions and offer advice intended to increase the effectiveness of the learning process. This is accomplished by requiring users to select the type of contribution (e.g. proposal, question or comment) from a list each time they add to the discussion. The system rates the collaboration between pairs of students along four dimensions: initiative, creativity, elaboration and conformity. These attributes, along with others such as the length of contributions, factor into a fuzzy reference procedure that rates students' collaboration on a scale from "awful" to "very good". The advisor elaborates on the attribute values and offers students tips on improving their interaction.

5.4.1.8 BetterBlether

BetterBlether (Robertson, Good and Pain, 1998) is a computer mediated communication environment for use in primary school classrooms. It provides structured support for children developing their communication skills in unsupervised group discussion.

The system can be used to discuss any topic. When pupils start a session with BetterBlether, the first question is displayed on the screen to stimulate their ideas and opinions about the topic. When the group feels that has fully covered the question, they have the option of moving on to another of the questions. The user can send messages to a particular group member or to the group as a whole. The message is constructed from one of the colored sentence openers buttons followed by additional text. The sentence openers are arranged and colored by skill category, but names of the skill categories are not stated. The skill categories are not taught explicitly, because some sentence openers could fit into more than one category.

All discussion contributions are logged to a text file, which can then be reviewed by the teacher or by the group members themselves.

5.4.1.9 Soller's system

Soller (1999) describes a collaborative learning system designed to teach group of engineers how to work together on software design problems. The users use a task window to construct, and can share these diagrams with their group members.

The system combines speech acts and sentence openers in a structured interface to help students provide effective help to their peers and to encourage them to engage in active learning. Sentence openers are grouped by speech acts types in the interface. The interface also displays logs of the numbers of each type of speech act used in the conversation and the number of contributions from each group member. The idea behind this display is that when students have access to information of this type, they are in better position to diagnose and repair problems in the group interactions. Further more, it may encourage students to reflect on their own contributions to the group and make an extra effort to improve weak skills.

5.4.1.10 Group Leader Tutor

'Group Leader Tutor' (McManus and Aiken, 1995) is intended to promote the collaboration skills identified in Johnson and Johnson (1994) during the course of problem solving discussions between two students. The students send messages to each other by selecting a sentence opener from a menu and then elaborating on this opener with additional text. An intelligent tutoring system offered advice and feedback on the student's skill use during the course of the discussion, and generated feedback at the end of the discussion. The tutoring system's suggestions are based on the concept that a conversation can be understood as a series of conversational acts (e.g. Request, Mediate) that correspond to users' intentions. Unlike other systems (e.g. 'Coordinator' by Flores *et al.*, 1988), users are not restricted to using certain acts based on system's beliefs. Group Leader monitors contributions from the students and compares them to an ideal model of interaction. This might be problematic because a system which uses no natural language understanding is not in an ideal position to criticize students' discussion skills through analysis of sentence openers.

Table 5.1 presents a summary of the above information about the reviewed CSCL systems.

SYSTEMS / purpose	**Pre-collaborative stage**	
	Age	**Task characteristics**
1 DIALAB / To teach argument and critical thinking	Students	Rigid logic-based dialogue game
2 CoVis / To transform science learning to better resemble the authentic practice of science	High school	Open-ended inquiry in science learning
3 Belvedere / To teach collaborative inquiry	9-12th grade	Scientific inquiry, investigation of real-world "challenge problems"
4 Knowledge Forum / To build your community Knowledge	Schools (From 4th grade students), Universities, for everyone	Creation of a Knowledge building community.
5 COLER / To solve database-modeling problems.	Students that have the right level of domain knowledge for using the system.	Database-modeling.
6 C-CHENE / To teach modeling and the concept of energy in physics.	Students (16-17 years old).	Modeling in physics.
7 BetterBlether / To develop communication skills in unsupervised group discussion.	Primary school classrooms.	
8 Group Leader Tutor / To promote collaboration skills during the course of problem solving discussions.	Students	
9 DEGREE/ To increase the effectiveness of the learning process.	Students	
10 Soller's / To teach a group of engineers how to work together on software design problems.	Adults	Object-oriented design problems.

Table 5.1: CSCL systems and their characteristics

5.4.2 Means of action and communication

5.4.2.1 Mode of communication

Collaborative learning is one of the pedagogical methods that can stimulate students to discuss information and problems from different perspectives, and to elaborate and refine these in order to re- and co-construct (new) knowledge or to solve the problems (Veerman, 2000). In such situations, externalization, articulation, argumentation, negotiation of multiple perspectives, are considered being the main mechanisms that can promote collaborative learning (Dillenbourg and Schneider, 1995; Baker, 1996; Veerman, 2000). So, collaborative learning requests interactions during the learning process for accomplishing the common goals but also for the participants to have the feeling that they belong at the same learning community. The participants in a CSCL system should be able to send and receive messages, and the system should minimize the effort of the participants to produce a message. There are **synchronous** collaboration environments which require students and instructors to be consistently connected and in constant attendance, and **asynchronous** collaboration environments which allow parties to communicate in a disconnected fashion and eradicate barriers of time and/or place. There are systems that support both as is shown in Table 5.2.

5.4.2.2 Structure of communication

A. Structure of dialogues.

In most Computer Supported Collaborative Learning Environments the communication is text-based, there is not a face-to-face communication and hence there is an increased risk of misinterpretation (Moore, 1993). Hence, it is not sufficient to provide distance learners with a communication channel. As we have already mentioned, the nature of the discussion varies considerably depending on whether the group is supervised or unsupervised: the presence of a teacher seems to encourage group members to justify their opinions and elaborate on their comments (Robertson, Good and Pain, 1998). However, a teacher is likely to dominate the discussion. A computer-mediated communication environment might bridge between supervised and unsupervised group work. The environment should provide some support and structure in an otherwise unsupervised group discussion.

Structuring the dialogue in a CSCL system:

1. Improve shared understanding by making explicit the (underlying) goal of an utterance.
2. Increase task-oriented behaviour and decrease off-task behavior (Baker and Lund, 1997), so students more easily focus on specific parts of the problem-solving process.
3. Improve students' collaborative argumentation (Veerman, 2000). By using a defined set of discourse acts and sentence openers students can be encouraged into certain discourse acts and problem-solving activities.

Structuring the dialogues can be done using **turn-taking**, or **sentence openers**, or **discourse acts**. Some of the Computer Supported Collaborative Learning Environments, like 'Group Leader Tutor' (McManus and Aiken, 1995), 'BetterBlether' (Robertson *et al.*, 1998) and 'Soller' (1999), structure the dialogues between the students during problem solving discussions. This structuring is based on others' work, like 'Johnson & Johnson' (Robertson, Good and Pain, 1998) and 'Wegerif & Mercer' (Pilkington, 2001). They have identified a series of skills and associated subskills that characterize effective group discussions. So, the students send messages to each other by selecting a sentence opener from a menu and then elaborating on this opener with additional text. There is a one—to-one correspondence between sentence openers and the skills identified. Others, like 'C-CHENE' (Baker and Lund, 1996), structure the dialogue on the basis of analysis of a corpus of ‘chat-box’ interactions, existing models for information dialogues (Moeschler, 1985; Bunt, 1995) and collaborative problem solving interactions (Baker, 1994).

This approach is reflected in Dillenbourg’s (1999) view, when he states that:

“One should not talk about the effects of collaborative learning in general, but more specifically about the effects of particular categories of interaction”.

There are some systems (DIALAB) that use **turn-taking** in order to ensure that all the learners will participate. But turn-taking control in the communication window may inhibit elaboration.

We can assume that there are advantages when the dialogue is structured. But there are objections that we must have in mind when we design CSCL environments:

1. A lot of people believe that if the participants of a collaborative learning situation could choose between a structured communication mode and a ‘free’ communication mode they would definitely choose the latter. Some experiments, (Jermann, 1999; Baker and Lund, 1996), have shown that the structured section of the interface was more frequently used than the free section. It appears that the usage of the free versus structured interface is not independent from the type of content being uttered. The management is more often expressed by using the free section while the task and strategy contributions are more often expressed by the structured one. Furthermore, the results show that pairs, who use the ‘free’ communication mode more than the ‘structured’ one, produce more ‘off-task’ statements than the pairs who preferred the ‘structured’ mode.
2. Requiring learners to select a sentence opener before typing the remainder of their contribution may tempt them to change the meaning of the contribution to “fit” one of the openers, thus changing the nature of the collaborative interaction. For this reason, it is critical that the openers enable the widest possible range of communications with respect to the learning task (Soller, 1999). The openers must originally develop from an analysis of face-to-face collaborative learning of the same task and environment, and should be refined several times to accommodate users feedback and observed experiences.
3. The sentence openers are not always used as intended, with the result that the contribution following the opener would not necessarily correspond to the discussion skill represented by the sentence opener. This is something that we must have in mind if we analyze the data manually (like ‘BetterBlether’).

B. Structure of the common workspace

In collaborative learning a shared final product is expected from the participants. So, in a CSCL system a common workspace must exist. In this way the learners have a shared point of reference, better mutual understanding and fulfillment of the shared goal. Of course all the participants must have access to the common workspace. As a result the production of the final product must be coordinated.

The systems adopt various ways for regulating the access. The *Action Key* (the student that poses the Action Key controls the common workspace) is a good choice (COLER), but we must consider the case of a possible deadlock, if the student does not give the Action Key to other learners that ask for it.

We must also consider the case in which the learners do not agree about an object that has been inserted at the common workspace. What about if the author does not agree to delete it?

In the task window graphic dialogue acts (Belvedere) improve consideration of multiple viewpoints and elaboration.

Table 5.2 A summary of CSCL systems about the means and the structure of communication.

SYSTEMS / Purpose	Means of action and communication				Structure of Communication (on dialogue and action)	
	Mode of communication	**Action**	**Private workspace**	**Common workspace**	**Structure of dialogues**	**Structure of the common workspace**
DIALAB / To teach argument and critical thinking	Synchronous	No	No	No	1. Sentence openers. 2. Set of dialogue rules (which determines which move type can follow each other). 3. Turn-taking (one dialogue move per turn).	There is not a common workspace.
CoVis / To transform science learning to better resemble the authentic practice of science	Synchronous and asynchronous	Use of Scientific Visualization Environment	Yes	Yes	1. Set of page types that provide both a description of their contents, and of their relationship to other pages. 2. Each page has links to other pages that inspired it, and links from the displayed page to other responses. 3. The link types that a page may have are determined by the type of that page.	
Belvedere / To teach collaborative inquiry	Synchronous and asynchronous	Construction of inquiry diagrams	Yes	Yes	Chat facility.	1. Discourse acts. 2. Diagrams. Discourse acts and evidential relations (Each statement has a type, data, hypothesis, unspecified, and at least one link, against, for, to another statement). 3. Only one person can work at an item at a time.
Knowledge Forum / To build your community Knowledge	Synchronous and asynchronous	Building databases	Yes	Yes	1. Sentence openers. 2. Idea networks.	The author can delete a node.
COLER / To solve database-modeling problems.	Synchronous	Construction of ER diagrams	Yes	Yes	No structure, only a chat facility	Action Key

C-CHENE / To teach modeling and the concept of energy in physics.	Synchronous	Construction of energy chains	No	Yes	1. Set of communicative acts, grouped according to their function. 2. Strict turn-taking (with possibility of interruption).	At any time, only one student can 'act', (with possibility of interruption).
BetterBlether / To develop communication skills in unsupervised group discussion.	Synchronous	No	No	No	Sentence openers.	There is not a common workspace.
Group Leader Tutor/ To promote collaboration skills during the course of problem solving discussions.	Synchronous	No	No	No	Sentence openers.	
DEGREE / To increase the effectiveness of the learning process.	Asynchronous	No	No	No	Discourse acts.	There is not a common workspace
Soller's / To teach a group of engineers how to work together on software design problems.	Synchronous	Yes OMS software design diagrams	No	Yes	Sentence openers.	

5.4.3 Awareness

Goldman (1992) identifies three types of awareness: social, task and conceptual. Social awareness is the awareness that students have about the social connections within the group (What should I expect from other members of this group? How will I interact with this group? What role will I take in this group? What roles will the other members of the group assume?). *Social awareness* is inter-personal and perhaps best supported implicitly. For example, audio/video conferencing can create communication opportunities that let people exchange necessary information with each other and negotiate their roles. *Task awareness* is the awareness of how the task will be completed (What do I know about this topic and the structure of the task? What do other know about this topic and the task? What steps must we take to complete the task? What tools are needed to complete the task? How much time is required? How much time is available?). *Concept awareness* is the awareness of how a particular activity or piece of knowledge fits into the student's existing knowledge (How does this task fit into what I already know about the concept? What else do I need to find out about this topic? Do I need to revise any of my current ideas in light of this new information?). Support for both task and concept awareness has been considered in collaborative learning and CSCL research; this support often provides explicit structures that students can use as scaffolds to assist them in organization or to help them stay focused on the learning tasks.

Additionally there is the *workspace awareness:* the up to the minute knowledge a student requires about other students interactions with the shared workspace (What are the other members of the group doing to complete the task? Where are they? What are they doing? What they have already done? How can I help other students to complete the project?). This awareness is essential if the students are to learn and work together effectively (Gutwin, Stark and Greenberg, 1995). Collaborating learners maintain this awareness by tracking information such as other learners' locations in the shared workspace, their actions, the interaction history. Workspace awareness is necessary for effective collaborative work, but also plays an integral part in how well an environment creates opportunities for collaborative learning. This awareness is important in collaborative learning for two reasons. First, it reduces the overhead of working together, allowing learners to interact more naturally and more effectively. Second, it enables learners to engage in the practices that allow collaborative learning to occur.

So the system must display information such as: who is participating; who is controlling the share workspace; what is the contribution to the final object for each student; what each student is doing (e.g. typing, waiting for the Action key, etc.). Visualization structures of students' discussion and actions with the aid of a suitable representation can assist students be aware of other's actions and opinions. For example a graphical overview of the discussion on-line facilitates the students to keep track of the discussion (Nakamura, Hanamoto, and Otsuki, 1999; Veerman and Treasure-Jones, 1999).

Table 5.3 A summary of CSCL systems about awareness.

SYSTEMS / purpose	**Awareness**		
	Learners' actions	**Historic of actions**	**Historic of dialogues**
DIALAB / To teach argument and critical thinking	What You See Is What I See		Each participant's statements are visible to both participants.
CoVis / To transform science learning to better resemble the authentic practice of science	What You See Is What I See		1. Each page has links to other pages that inspired it. 2. Displays an overview for each notebook's pages, including their types, titles and relational structure.
Belvedere / To teach collaborative inquiry	What You See Is What I See. A *panel* shows which teammates are already connected.	1. Optional entry of the names of the learners that "wrote" each node. 2. The inquiry diagram serves as a record of what the students have done.	Chat history
Knowledge Forum / To build your community Knowledge	What You See Is What I See.	1. Register of the names of the learners that "wrote" each node. 2. The nodes can be sorted by date, thread or author. 3. Search functions allows for the bringing together of nodes of interest to the searcher.	
COLER / To solve database-modeling problems.	What You See Is What I See. *Team panel:* shows which teammates are already connected. O*pinion panel:* shows teammates ' opinions on a current issue. *Floor control panel:* shows the name of the student who has the control of the common workspace, and the students waiting for a turn.	Register of the names of the learners that "wrote" each contribution.	Chat history.
C-CHENE/ To teach modeling and the concept of energy in physics.	What You See Is What I See.	All actions are added, numbered and time-stamped at the interaction history window.	All dialogues are added, numbered and time-stamped at the interaction history window.

BetterBlether / To develop communication skills in unsupervised group discussion.			History of the conversation
Group Leader Tutor/ To promote collaboration skills during the course of problem solving discussions.			History of the conversation
DEGREE / To increase the effectiveness of the learning process.			History of the conversation
Soller's *et al.* / To teach a group of engineers how to work together on software design problems.	What You See Is What I See.		History of the conversation

5.4.4 Teacher's support.

It has been shown that children benefit from group discussions if the class teacher takes an active part in the group interaction (Harwood, 1995). Harwood observed that children as young as eight working in unsupervised groups were unlikely to use questioning and listening skills during the discussion, their discussions tended to lack continuity and they experienced problems with group relations. The participation of the teacher prompted group members to elaborate on and justify their opinions. Harwood (quoted in Robertson, Good and Pain, 1998) acknowledges that children need the experience of working in an unsupervised group so that they can develop for themselves the skills of managing group relations. Further analysis of Harwood's results (Robertson, Good and Pain, 1998) suggests that the presence of the teacher decreases the opportunities for children to take the initiative in the discussion, or to ask questions. A teacher is likely to dominate the discussion. Pupils in unsupervised groups were more likely to initiate their own ideas, move the conversation onto a new topic, and interact with their peers. Perhaps they felt that the responsibility for these sorts of interactions lay with the teacher.

So, in the first stages of curriculum education teachers are expected to spend time interacting with small groups encouraging and supporting the pupils, while in the more advanced stages of curriculum education, group of pupils work independently of the teacher. Thus, the teacher's role moves progressively from one of the group leader, to one of facilitator, where s/he provides scaffolding for the discussion but does not take an active role, and finally, to one of the observer. But observation of several groups, taking place in parallel, is not possible.

So:

1. A record of the discussion that has taken place (which could be provided for reference);
2. information about the task each learner works on, attendance of every learner's private workspace, attendance of the public workspace, elaborated comparative information concerning various workspaces (e.g presenting interesting differences); and
3. an elaborated analysis (historic) of the student's actions in the shared workspace, what each student has offered/done

would allow the teacher to detect when particular children are having difficulties; to identify skills which seem to be generally weak across a group of children and may also suggest when a group of children are not working well together. This might lead to the identification of skills to be addressed through teaching or prompting in teacher supported discussion, or to giving the whole class further teaching on skills identified as weak. It may also provide material for further analysis of group dynamics by the teacher.

Table 5.4 A summary of CSCL systems about teacher's support.

SYSTEMS / purpose	Teacher's support
DIALAB / To teach argument and critical thinking	
CoVis / To transform science learning to better resemble the authentic practice of science	There is a record with the history of the whole process, available only to the teacher.
Belvedere/ To teach collaborative inquiry	Student-constructed diagrams provide the teacher with a basis for assessing students' understanding of scientific inquiry, as well as of subject matter Knowledge.
Knowledge Forum / To build your community Knowledge	
COLER / To solve database-modeling problems.	Documents to describe the chronological sequence of events of the collaborative session in reference to a specific student, and the current state of the environment associated with each event. It also includes the existence of chat contributions, but not the exact words.
C-CHENE / To teach modeling and the concept of energy in physics.	
BetterBlether / To develop communication skills in unsupervised group discussion.	All discussion contributions are logged to a text file
Group Leader Tutor / To promote collaboration skills during the course of problem solving discussions.	
DEGREE / To increase the effectiveness of the learning process.	
Soller's / To teach a group of engineers how to work together on software design problems.	1. History of dialogues. 2. The final shared product.

5.4.5 Guidance of collaborative learning interactions.

According to Jermann (1999) we can classify the computer supported collaborative learning systems into two categories according to who is managing the interactions between the participants: the participants themselves or the system.

1. In the first case the learners are expected to manage the interaction having been given some kind of information about the interaction. So,
    a. There are systems that reflect the actions (**mirroring systems**). They gather data about the students' interaction, and display this information to the user without any further elaboration. For example, they display a history of the dialogues, or make the students/teachers aware of the participants' actions. The data displayed to the students do not undergo any processing but directly reflects the action taken on the interface. It is then up to the students to

interpret the situation and decide what actions (if any) to take. Belvedere, Covis, CSILE, C-CHENE, BetterBlether etc. belong to this category.

b. There are systems that model the state of interaction via a set of indicators that are displayed to the users (**monitoring systems**). Of course it is up to the user to interpret this information. These systems focus on quantitative aspects of interaction, like the number of messages, the number of problem solving actions, using different ways of visualization (Soller, 1999; Jermann, 1999; Simoff, 1999; Ogata, Matsuura, and Yano, 2000).

c. There are systems that interpret the content of the interaction (**monitoring systems**). The interpretation of actions by the system is facilitated through a structured interface. This information is either intended to be used later by a coaching agent (EPSILON), or to be used by the system in order to decide when to stop the interaction (Dialab).

2. In the second category the system is expected to manage the interaction between the users (**advising systems)**. The system is responsible for guiding the students toward effective collaboration and learning. It offers advices and is expected to play a role similar to that of a teacher in a collaborative learning classroom. There are systems that:

   a. Analyze the groups' conversation in order to decide the quality of the interaction (DEGREE, Group Leader Tutor).

   b. Analyze the actions of the participants in order to decide when to mediate (COLER).

Mirroring systems record and reflect input data. The most basic level of guidance that a system might offer involves making the students or the teachers aware of the participants' actions. Actions taken on shared resources, or those that take place in private areas of a workshop may not be directly visible to the collaborators, yet may significantly influence the collaboration. Raising awareness about such actions may help students maintain a representation of their teammates' activities.

Monitoring and advising systems process this input data to obtain a representation which is either displayed to the collaborators or used by the system. This representation may be quantitative or qualitative in nature. A quantitative derivation process might entail counting, for instance, the number of dialog or workspace actions a user has taken. A qualitative derivation process requires taking relational information into account, such as interdependencies between actions or between actions and application context.

Table 5.5 A summary of CSCL systems about guidance of learning interactions.

SYSTEMS / purpose	**Input data**	**Management of the interaction**	**Intervention Type**
DIALAB / To teach argument and critical thinking	Messages	By the students	Monitoring
CoVis / To transform science learning to better resemble the authentic practice of science	Messages	By the students	Mirroring
Belvedere/ To teach collaborative inquiry	Dialog, shared and private actions	By the students	Mirroring
Knowledge Forum / To build your community Knowledge	Messages, idea networks	By the students	Mirroring
COLER / To solve database-modeling problems.	Shared and private actions, dialog	By the system	Advising
C-CHENE/ To teach modeling and the concept of energy in physics.	Messages, shared actions	By the students	Mirroring
BetterBlether/ To develop communication skills in unsupervised group discussion.	Messages	By the students	Mirroring
Group Leader Tutor/ to promote collaboration skills during the course of problem solving discussions.	Messages	By the system	Advising
DEGREE/ To increase the effectiveness of the learning process.	Messages	By the system	Advising
Soller's / To teach a group of engineers how to work together on software design problems.	Messages	By the students	Monitoring

5.5 Conclusions and Implications for User Requirements.

Having in mind what we have mentioned up to now and based on the dimensions of collaborative learning identified by Kumar (1996), we outline some elements that should be addressed in the design, development and use of CSCL systems:

1. **Pre-collaborative stage.** Before starting a collaborative learning session in order to ensure reliability and effectiveness of collaborative learning, it is essential to identify the tasks that are to be satisfied. Learning is more likely to occur in certain educational situations. Optimal tasks should be open-ended. These types of

problems are characterized by the existence of justifiable beliefs and multiple acceptable viewpoints (Baker, 1992). Thus, students can share and learn from each other's differences in prior knowledge, experiences, beliefs and values.

Identifying peers and verifying their suitability for collaboration is another pre-collaboration task. The peers should be available over the computer network that supports collaboration. Also it is needed to consider the number or group size of collaborating peers. Most experiments are conducted with small numbers of peers to reduce complexity.

2. **Control over collaborative interactions.** Collaborative learning requests interactions during the learning process for accomplishing the common goals but also for the participants to have the feeling that they belong at the same learning community. The participants in a CSCL system should be able to send and receive messages, and the system should minimize the effort of the participants to produce a message. But the system must not act simply as a vehicle for collaboration. It must be involved in controlling the collaboration, facilitates the negotiation and the whole process of problem solving:

   a. **Structuring the dialogue** will improve shared understanding by making explicit the underlying goal of an utterance, increase task-oriented behaviour so students more easily focus on specific parts of the problem-solving process, improve students' collaborative argumentation because they are encouraged to use certain discourse acts and problem-solving activities. Structuring the dialogues can be done using turn-taking (DIALAB) or sentence openers (Soller, BetterBlether C-CHENE), or discourse acts (CoVis). Turn-taking control in the communication window may inhibit elaboration, while discourse acts may easily not be used as intended, with the result the contribution following would not necessarily correspond to the discussion skill represented by the discourse act. Also, requiring learners to select a sentence opener before typing the remainder of their contribution may tempt them to change the meaning of the contribution to "fit" one of the openers, thus changing the nature of the collaborative interaction. For this reason, it is critical that the openers enable the widest possible range of communications with respect to the learning task (Soller, 1999). The openers must originally develop from an analysis of face-to-face collaborative learning of the same task and environment, and should be refined several times to accommodate users feedback and observed experiences.

   b. **Structuring public workspaces**.

      In collaborative learning a shared final product is expected from the participants. So, in a CSCL system a common workspace must exist. In this way the learners have a shared point of reference, better mutual understanding and fulfillment of the shared goal. Of course all the participants must have access to the common workspace. As a result the production of the final product must be coordinated.

      The systems adopt various ways for regulating the access. The *Action Key* (the student that poses the Action Key controls the common workspace) is a good choice (COLER), but we must consider the case of a possible deadlock, if the student does not give the Action Key to other learners that ask for it. We must also consider the case in which the learners do not agree about an object that has been inserted at the common workspace. What about if the author does not agree to delete it? In the task window graphic dialogue acts (Belvedere) improve consideration of multiple viewpoints and elaboration.

   c. **Guidance of collaborative learning interactions.**

      In computer supported collaborative learning situations the systems must manage the interactions between the participants:

i. By recording and reflecting the actions of the learners (*mirroring systems)*. The most basic level of guidance that a system might offer involves making the students or the teachers aware of the participants' actions (Belvedere, CoVis, CSILE, C-CHENE). Actions taken on shared resources, or those that take place in private areas of a workshop may not be directly visible to the collaborators, yet may significantly influence the collaboration. Raising awareness about such actions may help students maintain a representation of their teammates' activities. It is then up to the participants to interpret the situation and decide what actions (if any) to take.

ii. By processing the input data to obtain a representation (*monitoring systems)* which is either displayed to the participants and is up to them to interpret this information (Soller, 1999), or is interpreted by the system for later usage by a coaching agent (EPSILON), or is interpreted by the system in order to decide when to stop the interaction (DIALAB).

iii. By processing the input data to obtain a representation which is used by the system to manage the interactions between the participants (*advising systems)*. The system is responsible for guiding the students toward effective collaboration and learning. It offers advices and is expected to play a role similar to that of a teacher in a collaborative learning classroom. There are systems that analyze the groups' conversation in order to decide the quality of the interaction (DEGREE, Group Leader Tutor), and others that analyze the actions of the participants in order to decide when to mediate (COLER).

So, monitoring and advising systems process this input data to obtain a representation which is either displayed to the collaborators or used by the system. This representation may be quantitative or qualitative in nature. A quantitative derivation process might entail counting, for instance, the number of dialog or workspace actions a user has taken. A qualitative derivation process requires taking relational information into account, such as interdependencies between actions or between actions and application context.

3. **Coordination of collaboration.** If we want students to be able to work collaboratively, they must be aware about other participants' actions. Collaborating learners maintain this awareness by tracking information such as other learners' locations in the shared workspace, their actions, the interaction history. Awareness is necessary for effective collaborative work, but also plays an integral part in how well an environment creates opportunities for collaborative learning. Awareness is important in collaborative learning for two reasons. First, it reduces the overhead of working together, allowing learners to interact more naturally and more effectively. Second, it enables learners to engage in the practices that allow collaborative learning to occur. So the CSCL systems must display information such as: who is participating(Belvedere, COLER); who is controlling the share workspace(COLER); what is the contribution to the final object for each student (Belvedere, COLER, Knowledge Forum); What each student is doing e.g. typing, waiting for the Action key, etc.(COLER). Visualization structures of students' discussion and actions with the aid of a suitable representation can assist students be aware of other's actions and opinions. For example a graphical overview of the discussion on-line facilitates the students to keep track of the discussion (Nakamura, Hanamoto, and Otsuki, 1999; Veerman and Treasure-Jones, 1999).

4. **Teacher's support.** In the first stages of curriculum education teachers are expected to spend time interacting with small groups encouraging and supporting the pupils, while in the more advanced stages of curriculum education, group of pupils work independently of the teacher. Thus, the teacher's role moves progressively from one of the group leader, to one of facilitator, where s/he provides

scaffolding for the discussion but does not take an active role, and finally, to one of the observer. But observation of several groups, taking place in parallel, is not possible. Most of the CSCL systems provide a record of the discussion that has taken place and an elaborated analysis (historic) of the students' actions in the shared workspace (CoVis, Belvedere, COLER). In order to allow teacher to detect when particular children are having difficulties, to identify skills which seem to be generally weak across a group of children and also to be able to suggest when a group of children are not working well together, the system must provide information about the task each learner works on, attendance of every learner's private workspace, attendance of the public workspace, elaborated comparative information concerning various workspaces (e.g presenting interesting differences).

6. The previous system MODELSCREATOR and users' requirements for its extension and improvement

6.1 Model Creator History

This project was funded by the Hellenic Minister of Education, in the frame of the program "Seirines" (1998-1999), while a further funding was provided as an extension of that project in the frame of the "*Penelope"* Program (2000-2001).

6.2 Main components and functionality of MODELSCREATOR

The MODELSCREATOR, as described in its accompanying manuals consists of four main components (figure 6.1): the space of "Study Themes", the "Modelling Space", the "Notebook" and the "Encyclopaedia of Models". It also contains a "Communication system", a "File Management system" and a "Help system".

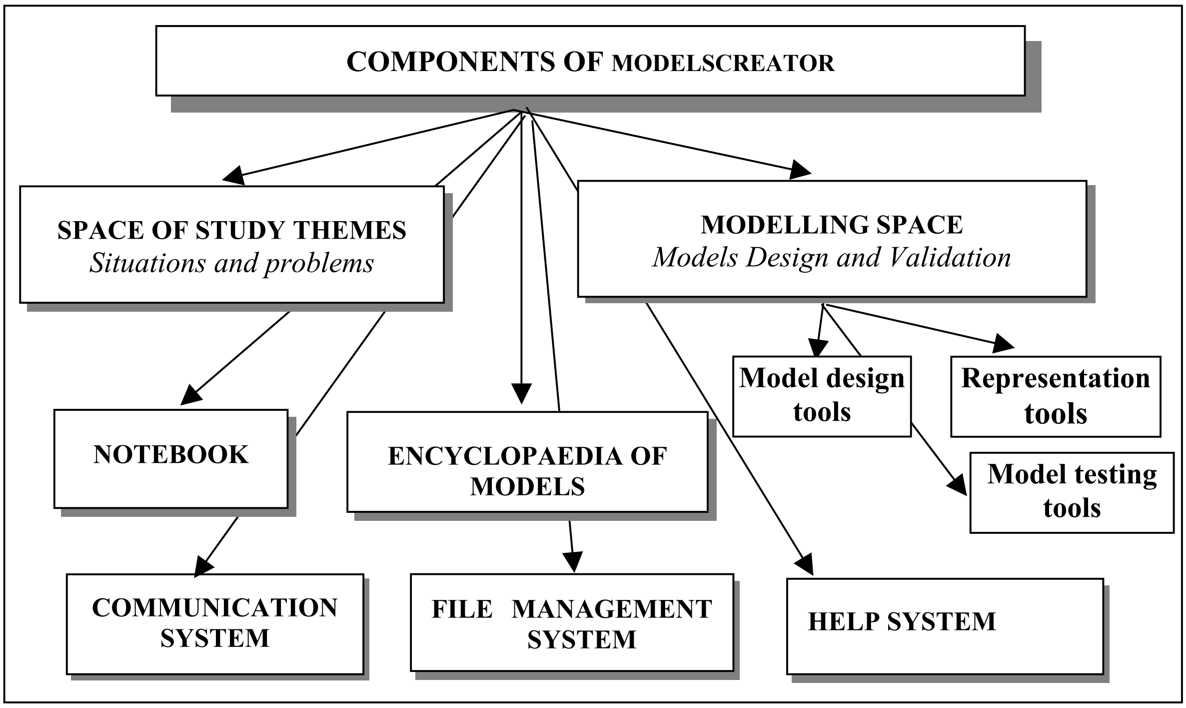


Figure 6.1: The architecture of MODELSCREATOR

1. The component *"Study Themes"* contains a number of situations that are proposed for modelling, while it allows for the creation of new problems by teachers and students. This special editor allows the presentation of new situations or phenomena by writing and formatting texts and by inserting various multimedia elements such as images and photos.

2. The "*Modelling Space*" permits the design, testing and validation of models. It consists of the area where the models can be designed. It contains the tools available to design them, the representation tools and the tools that are necessary to run the model. In order to design a model, students have to determine the model' s entities, their properties and the relations between them. The model design tools contain the lists of entities (left of screen, fig. 6.2) and the lists of relations (right of screen, fig. 6.2). There are two kinds of entities: the concrete ones (that correspond to objects) and the abstract

ones (that may correspond directly to concepts). The designation of each entity either requires both the iconic and the textual determination which are able to specify the concept, or the variable referring to this object or only the textual determination which directly refers to a concept. For each entity, one or more variables or concepts have to be determined as well as the estimation of their values (if necessary). The effect of these values appears directly in a visual mode. For instance, in the semi-quantitative reasoning mode, the fact that the value of the volume of the water (variable) in a barrel (object) is high or low can be seen from the water level in the icon of the barrel.

The students can choose and determine the desirable relation between two entities among the available relations of the four great categories:

- *Qualitative logical relations*: these relations can be expressed by logical operators (AND, OR, NOT) and control conditions (if…then), useful in decision making (see Figure 6.3).
- *Qualitative semantic relations*: they are able to produce concept maps. The concept maps are particularly useful to present and study the relations between concepts in various subjects (environmental education, physics, history, etc).
- *Semi-quantitative relations*: these relations are in terms of variation of properties' values and direction of this variation. In the current version, the student can use simple relations that correspond to simple algebraic relations which are common in mathematics and physics. Each relation is represented by a symbol. For instance, the relations of analogy or inverse analogy are expressed through the reasoning "If the one entity increases, the other one might increase, decrease, or remain unchanged" and are represented by special graphic symbols (↑↑,↑↓,↑↔,↑↑, or ↑↓).

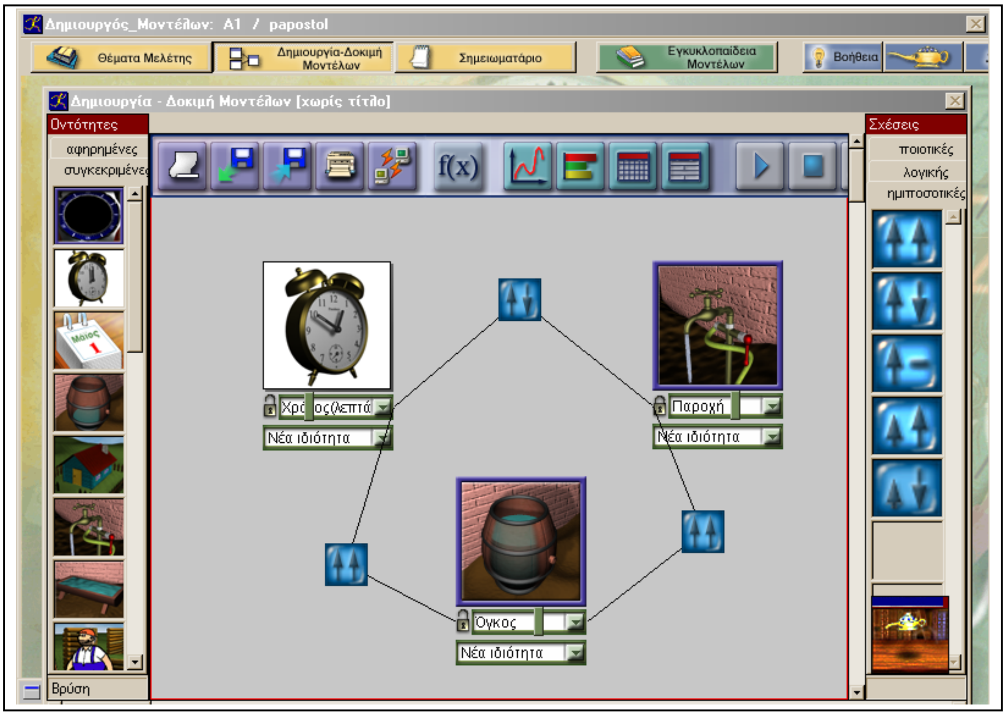


Figure 6.2: The Modelling_Area of MODELSCREATOR

- *Quantitative simple algebraic relations*: they are defined by arithmetic operators (+, -, *, /, =). In a direct manner, a dialogue box permits to relate some of the variables - previously defined in a model - with arithmetic operations in order to determine a quantitative algebraic relation.

The student having created a model can now run a dynamic model during which the simulation of the modelled phenomenon appears in the area of entities' icons. For example, when the model that is presented in Figure 6.2 runs, the student can see the water filling up the barrel for as long as the tap is turned on. If the relation between the barrel's volume and tap's rate of flow is an inverse analogy, it will result in the decrease of the water's volume in the barrel while the tap is on. In the case of decision models (as in the example shown in Figure 6.3) the student can see -in the icon related to the effect ('then')- the simulation of the decision's consequences (a boy will either cross the street safely or will have an accident).

Before or after model running, the student can activate the "Representation Tools" which include: tables of data, graphs, bar-charts, and tables of decision, associated with the desirable entities or variables that are specified through a simple dialogue box.

3. The ***"****Notebook"*: stimulates students to take notes during the modelling process. The specially designed notebook contains three folders "determination of situation", "model designing", and "model interpretation" (see Figure 6.4), in order to encourage students to put their thoughts down during the initial study of the situation (determination of situation, objectives and usefulness of model), during the design of models (which factors are considered as important and which are negligible, what are their predictions), and finally during the test and validation of model (how the model behaves, how this behaviour can be interpreted, what are the differences from reality, etc).

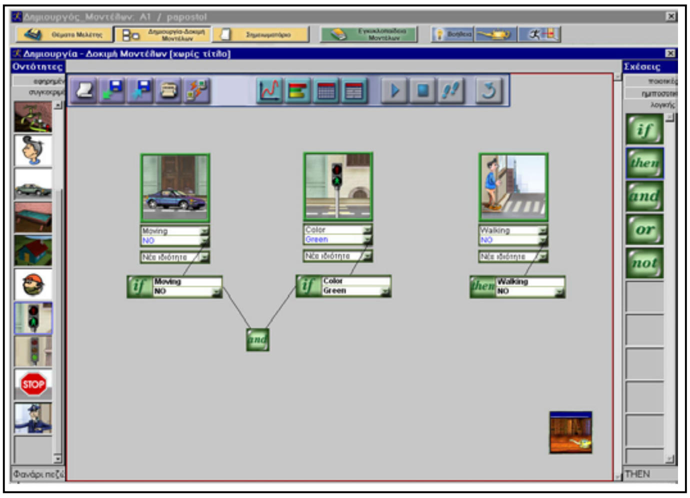


Figure 6.3: A decision making logical model

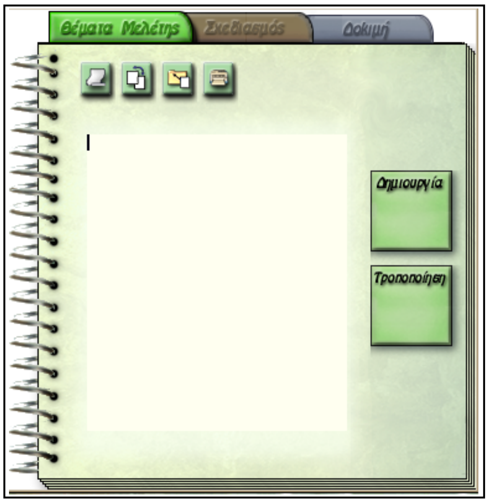


Figure 6.4: The special Notebook

4. The *"Models' Encyclopaedia":* constitutes a multimedia presentation, which aims to present and explain in a simple way the status and the nature of models and modelling processes in sciences and everyday life, using graphics, icons, video, narration and texts. This presentation has been developed because of the lack of any kind of appropriate reference sources related to the matter of models.

5. The *"Communication system"*: allows the exchange of problems, models, and notes (in an associated whole) between any users of MODELSCREATOR whether they are in a local area network (LAN) or in a distant Internet node. The exchange of data doesn't require the user to invoke a third party application (mail/ftp/etc. clients) since MODELSCREATOR works as both a server and a client for such type of transactions using protocol TCP/IP internally. It requires that the distant receiver is on-line, and there is not the possibility of a common collaborative work.

6. The *"Files' management" system:* implements a customised and personalised view of the file system hiding the file's real location on the hard disk and showing the hierarchy of the classes and their students/users instead. In addition, a special tree is 'assigned' to every user (student or teacher) which represents the various themes of study and the underlying created models. As a result, all a user needs to know to access their created models is their name in the system, the class they belong to, and the specific subject of study they are interested in. The real location of the files on the disk is invisible to them,

protecting them from dangers that direct access to the underlying file system inherently creates, especially when given to inexperienced users.

7. The *"Help system"*: It contains two sub-components, an on-line help (a hypertext system), and an immediate context sensitive help that gives audio guidelines on using the main components.

6.3 Development approach

Most of the above components have been developed in Microsoft Visual C++ using the Win32 API (Application Programming Interface), while two components (the representation tools and the notebook) have been developed using MFC (Microsoft Foundation Class). The multimedia presentation of Models' Encyclopaedia has been developed in Macromedia Director.

More information concerning its development is presented in the report "Software requirements" DO6, pp.136-137.

The use of the MODELSCREATOR in school settings

In the context of the MODELSCREATOR, modelling takes different forms and can favour the exploratory as well as the expressive way of learning. During exploratory activities of learning the student examines ideas over a subject using prepared models, which have been expressed from someone else (other students or teachers). During expressive activities the student expresses his own ideas by producing new models. Actually, there are a number of proposed themes of study referring to mathematics, physics, environmental education and every day decision-making. Most of the proposed situations are created in order to treat themes where students present difficulties in the implicated concepts, i.e. the notion of analogy in mathematics, or the notion of acceleration and force in kinematics. These activities fit with the curricula in Greece, in Junior High School (Gymnasium in Greece) for students 12-15 years old.

An extended manual for teachers containing a user's guide, general guidelines of suitable pedagogical approaches and detailed guidelines about the proposed modelling activities, was produced. Teachers have used ModelsCreator in some experimental schools of the initiative of ICT integration in secondary schools called ODYSSEIA.

6.4 Requirements for extension and improvements

During experimental use in schools, some of these requirements were expressed by teacher –researchers, others by teachers working in schools with their students.

The main requirements that were expressed (provided through oral comments during panel interviews and written recommendations) are summarized as following, and internally distinguished in two different internal sub categories, as improvements and as extensions:

A. General requirements

Extensions needed

The functionality to communicate some models is too constraining and not sufficient to really support exchanges and collaborative activities between teachers or students working in various schools.

Improvements

- The system is considered as 'heavy', taking too much time to load.

- The interface of the system presents some inconsistencies that provoke confusion in experienced users of ICT applications (it is not fully compatible with known standards).

B. About models

Extensions

Some extensions of the functionalities related to models creation are needed. In particular:

- the possibility to create composite models;
- to model situations that evolve according to different models during different periods of time.

Improvements

- To improve the interface of relations used for concepts maps in order to become more usable.
- To improve the interface of algebraic relations in order to become more usable.
- To provide entities that are editable by young students.
- To reconsider which from the available semi-quantitative relations could be really expressed by students, and omit these that could not be expressed.
- To incorporate new entities, more abstract ones, that support the transition from object oriented entities to abstract and symbolic ones, such as: fill in object from left to right, fill in object from down to up, car moving, truck moving, quantity of people, clock, spring, etc., that will allow to create a wider range of new problems.

C. About representations of data

Improvements related to the 'graph' facility

- Provide zoom in-zoom-out functionality that is not general, but is specific to each axis.
- Provide new functionality that automatically fits the best scale.
- Provide a better interface to indicate the variables that will be presented by a graph.

D. About teachers support

Extensions

- During class work, when teachers have 10 or even 15 groups of two students working on computers, it is very difficult to follow and support them. It would be a valuable help, if somehow we could have some information on how students interact during activities in order to know how to guide and support them.

7. References and Bibliography

AGENTSHEETS web site: http://www.agentsheets.com

Anapolitanos, A.D. (1985) *Introduction in Mathematics Philosophy* (In the Greek language), 3rd ed., (Athens: NEFELI Publications).

Anderson-Inman, L. and Zeitz, L. (1993) Computer-based concept mapping: Active studying for active learners. *The Computing Teacher*, 21-(1), pp. 1-5.

Astolfi, J.P., Drouin, A.M. (1992) La modélisation à l'école élémentaire. In Equipe INRP/LIREST, *Enseignement et apprentissage de la modélisation en sciences*. INRP.

Ausubel, D.P. (1968) *Educational psychology: A cognitive view*. (Holt, Rinehart & Wilson).

Bachelard, S. (1979) Quelques aspects historiques des notions de modèle et de justification des modèles. In P. Delathre and M. Thellier (Eds) *Elaboration et justification des modèles.* Two volumes. (Paris: Maloine SA) Vol.1 pp9-20.

Bailey, J. (1996) *After thought: the computer challenge to human intelligence.* (New York: BasicBooks).

Baker, M. (1996). *Argumentation and cognitive change in collaborative problem-solving dialogues* (Report No. CR-13/96). (France: COAST).

Baker, M.J. and Lund, K. (1997). *Flexibly structuring the interaction in a CSCL environment*, COAST research team, UMR-GRIC, CNRS/University of Lyon.

Baron, G.L., Bruillard, E. and Dansac, Ch. (1999) *Représentations, modèles et modélisations; implications sur les stratégies éducatives et sur les processus d'apprentissage: synthèse bibliographique.* Representation Consortium, Project Ref. MM 1045, Deliverable 1, January 1999.

Barros, B. and Verdejo, F., M. (2000) Analyzing student interaction processes in order to improve collaboration. The DEGREE approach, *International Journal of Artificial Intelligence in Education, (2000),* 11, 221-241.

BECTa (2000a) *Curriculum Software Initiative Report: Geography.* (http://www.becta.org.uk/technology/software/curriculum/reports/geography.html)

BECTa (2000b) *Curriculum Software Initiative Report: Science.* (http://www.becta.org.uk/technology/software/curriculum/reports/science.html)

BECTa (2000c) *Curriculum Software Initiative Report: ICT.* (http://www.becta.org.uk/technology/software/curriculum/reports/ict.html)

BECTa website: http://www.becta.org.uk

Berlekamp, E.R., Conway, J.H., Guy, R.K. (1982) *Winning ways for your mathematical plays (Vol. 2).* (London: Academic Press).

Bertalanffy, Von L. (1969) *General system theory*, 2nd ed., (New York: G. Brazilier Inc).

Black, M. (1962) *Models and Metaphors: Studies in Language and Philosophy.* (New York: Cornell University Press).

Black, P.J. and Lucas, A.M. (1993) *Children's informal ideas in science.* (London: Routledge).

Bliss, J. (1994a) From mental models to modelling. In H. Mellar, J. Bliss, R. Boohan, J. Ogborn, and C. Tompsett (Eds.), *Learning with artificial worlds: computer based modeling in the curriculum.* (London: The Falmer Press). pp 27-32.

Bliss, J. (1994b) Reasoning with a semi-quantitative tool. In H. Mellar, J. Bliss, R. Boohan, J. Ogborn, and C. Tompsett (Eds.), *Learning with artificial worlds: computer based modeling in the curriculum.* (London: The Falmer Press). pp 128-141.

Bliss, J. (1994c) ‘Causality and common sense reasoning’. In H. Mellar, J. Bliss, R. Boohan, J. Ogborn, and C. Tompsett (Eds.), *Learning with artificial worlds: computer based modeling in the curriculum.* (London: The Falmer Press). pp 117-127.

Bliss, J. and Ogborn J. (1989) Tools for exploratory learning. *Journal of Computer Assisted Learning 5*(1), pp 37-50.

Bliss, J. and Saconidis, H. (1994) ‘Reasoning with a qualitative modelling tool’. In H. Mellar, J. Bliss, R. Boohan, J. Ogborn, and C. Tompsett (Eds.), *Learning with artificial worlds: computer based modeling in the curriculum.* (London: The Falmer Press). pp 153-161.

Bliss, J., Ogborn, J., Boohan, R., Briggs, J., Brosnan, T., Brough, D., Mellar, H., Miller, R., Nash, C., Rodgers, C., and Sakonidis, B. (1992). Reasoning Supported By Computational Tools. *Computers Education*, *18*(1-3), pp 1-9.

Bobrow, D.G. (1984) Qualitative Reasoning about Physical Systems: An Introduction, *Artificial Intelligence,* 24, pp 1-6.

Boohan R (1995) ‘Why is a rumour like an infectious disease? - computer modelling with WorldMaker’. Personal communication.

(http://www.richard-boohan.org/publications/WM95.doc)

Boohan, R. (1992) ‘WorldMaker: an object-based approach to computer modelling’, Proceedings of the European Conference about Information Technology in Education, University of Barcelona, November 1992, pp 30-39.

Boohan, R. (1994a) ‘Interpreting the world with numbers: An introduction to quantitative modelling’. In H. Mellar, J. Bliss, R. Boohan, J. Ogborn, and C. Tompsett (Eds.), *Learning with artificial worlds: computer based modeling in the curriculum.* (London: The Falmer Press). pp 49-58.

Boohan, R. (1994b) ‘Creating worlds from objects and events.’ In H. Mellar, J. Bliss, R. Boohan, J. Ogborn, and C. Tompsett (Eds.), *Learning with artificial worlds: computer based modeling in the curriculum.* (London: The Falmer Press). pp 171-179.

Boohan, R. and Brosnan, T. (1994) ‘Reasoning with a quantitative modelling tool’. In H. Mellar, J. Bliss, R. Boohan, J. Ogborn, and C. Tompsett (Eds.), *Learning with artificial worlds: computer based modeling in the curriculum.* (London: The Falmer Press). pp 59-67.

Boohan, R. and Maragoudaki, E. (1997) 'Computer worlds and the real world: exploring children's understanding with WorldMaker' Paper presented at ESERA Conference, Rome, September 1997

Bransford, D.J., Brown, L.A. and Cocking, R.R. (Eds), (2000*) How people learn. Brain, Mind, Experience and School*, (Washington, D.C.: National Academy Press).

Bratko, I. (1990) *PROLOG: Programming for artificial intelligence*, 2nd ed., (Addison-Welsey Inc)

Briggs, J. (1989) *Box Modeller.* Computer Porgram.

Brosnan, T. (1994) ‘Using spreadsheets to develop understanding in science’. In H. Mellar, J. Bliss, R. Boohan, J. Ogborn, and C. Tompsett (Eds.), *Learning with artificial worlds: computer based modeling in the curriculum.* (London: The Falmer Press). pp 76-84.

Brown, A.L. and Palincsar, A.S. (1989) Guided, cooperative learning and individual knowledge acquisition, In L. Resnick (Ed.), *Knowing, learning and instruction: Essays in Honor of Robert Glaser.* (pp. 393-451) (Hillsdale, NJ: Lawrence Erlbaum Associates).

Bruner, J. (1966) *Toward a theory of instruction.* (Harvard University Press).

Bunt, H.C. (1995). Dialogue Control Functions and Interaction Design. In R.J. Beun, M. Baker, M. Reiner (Eds.), *Dialogue and Instruction, Modeling Interaction in Intelligent Tutoring Systems,* Proceedings of the NATO Advanced Research Workshop on Natural

Dialogue and Interactive Student Modeling, (pp. 197-214). (Berlin, Germany: Springer).

Buzan T. and Buzan B. (1993) *The mind map book: How to use radiant thinking to maximize your brain's untapped potential*. (Plume Book).

Camara, A.S., Ferreira, F.C., Nobre, E. and Fialho, J.E. (1991) *The Picture Simulator.* (Uninova, Portugal: Newsoft Group).

Capozzi, M., Rothstein, P. and Curley, K. (1996) Approaches for distributed learning through computer-supported collaborative learning. Workshop conducted at the 1996 *Conference on Computer-Supported Cooperative Work (CSCW '96),* Cambridge, MA.

Chiu, C.H., Huang, C.C. and Chang, W.T. (2000) The evaluation and influence of interaction in network supported collaborative concept mapping. *Computers & Education*, Vol. 34, pp. 17-25.

Clocksin, W.F. and Mellish, C.S. (1981) *Programming in Prolog*, 1st ed., (Springer-Verlag).

COM (2002) *eEurope Benchmarking Report*. February 2002. Report 62. Brussels. (http://europa.eu.int/information_society/eeurope/news_library/new_documents/ /benchmarking/benchmarking_en.pdf)

Cooper, A. (1995) *About face, the essentials of user interface design.* (Foster City, CA: Programmers Press).

de Kleer, J., and Brown, J. S. (1981). 'Mental models of physical mechanisms and their acquisition'. In J. R. Anderson (Ed.) *Cognitive skills and their acquisition* (pp. 285-309). (Hillsdale, NJ: Lawrence Erlbaum).

de Kleer, J., and Brown, J. S. (1983) 'Assumptions and Ambiguities in Mechanistic Mental Models'. In D. Gentner and A. L. Stevens (Eds.) *Mental Models.* (Hillsdale, NJ: Lawrence Erlbaum).

Desautels, J. and Larochelle, M. (1992) *Autour de l'idée de science; itinéraires cognitifs d'étudiants et d'étudiantes.* (Bruxelles: De Boeck-Wesmael).

Dillenbourg, P. (1999) *Collaborative learning: Cognitive and computational approaches.* (Amsterdam: Pergamon).

Dillenbourg, P. and Schneider, D. (1995) Mediating the mechanisms which make collaborative learning sometimes effective. *International Journal of Educational Telecommunications, 1*(2-3), 131-146.

Dillenbourg, P., Baker, M., Blaye, A. and O'Malley, C. (1996) The evolution of research on collaborative learning. In E. Spada and P. Reiman (Eds) *Learning in Humans and Machine: Towards an interdisciplinary learning science.* (pp. 189-211). (Oxford: Elsevier)

Dillon, C. (1994). Qualitative Reasoning about Physical Systems - An Overview. *Studies in Science Education*, *23*, pp 39-57.

diSessa, A.A., Hoyles, C. *et al.* (Eds.) (1995) *Computers and Exploratory Learning*. (Berlin: Springer).

Doise, W., Mugny, G. and Perret-Clermon, A. (1975). Social interaction and the development of cognitive operations, *European Journal of Social Psychology, 5*(3), pp 367-383.

Donnely, D. (1996) *CIP's* Seventh Annual Educational Software Contest: The Winners: Modellus. *Computers in Physics, 10*(6), pp 536-537.

Driver, R., Guesne, E. and Tiberghien A. (1985) *Children's ideas in science.* (Milton Keynes: Open University Press).

Dumas Carré, A. and Weil-Barais, A. (Eds). (1998) *Tutelle et médiation dans l'éducation scientifique*. (Berne: Peter Lang).

Eigen, M. and Winkler, R. (1983) *Laws of the Game: How the Principles of Nature Govern Chance* (translated by Robert and Rita Kimber). (Harmondsworth, England: Penguin Books).

Einstein, A. and Infeld, L. (1938) *The Evolution of Physics: the growth of ideas from early concepts to relativity.* (Cambridge University Press).

Eisenberg, M. (1995) Creating software applications for children: some thoughts about design. In A. A. diSessa, C. Hoyles and R. Noss *Computers and exploratory learning*. (Berlin: Springer).

Engestrom, Y., Miettinen R. and Punamaki R.L. (1999) (Eds) *Perspectives on Activity Theory*. (Cambridge University Press).

Eurydice (2001) *Basic Indicators on the Incorporation of ICT into European Education Systems – Facts and figures – 2000/01 Annual Report.* (Brussels: Eurydice) (http://www.eurydice.org)

Feurzeig, W. (1993) Visualization in educational computer modelling. In T. de Jong, H. Spada and D. Tawne *The use of computer models for explication, analysis and experiential learning*. (Berlin, Springer-Verlag).

Fiolhais, C., Silva, L. *et al.* (1996) *Fvsica 12.̇I ano, Manual de actividades*. (Lisboa: Didαctica Editora).

Fisher, K. (1990) Semantic Networking: the new kid on the block*. Journal of Research of Science Teaching*, Vol. 27, No 10, pp. 1001-1018.

Fisher, K. (1992) SemNet: A Tool for Personal Knowledge Construction in Kommers P. *et al.* (Eds) *Cognitive Tools for Learning*. (Springer-Verlag).

Fisher, K., Wandersee, J. and Wideman, G. (2000) *Enhancing cognitive skills for meaningful understanding of domain specific knowledge,* American Association for the Advancement of Science (http://public.sdsu.edu/CRMSE/Fisher-one.com).

Flores, F., Graves, M., Hartfield, B. and Winograd, T. (1988) Computer systems and the design of organizational interaction. *ACM transactions on Office Information Systems, 6*(2), pp 153-172.

Forbus, K. (1984). Qualitative Process Theory. *Artificial Intelligence*, *24*(1-3), pp 85-168.

Forrester, J.M. (1968) *Principles of Systems.* (Cambridge, MA: Wright-Allen Press Inc.).

Franceschelli, S. and Weil-Barais, A. (1998) La construction d' un modèle en mécanique: analyse des interactions professeur-élèves. In A. Dumas Carré and A. Weil-Barais: *Tutelle et médiation dans l'éducation scientifique.* (pp. 211-238*)* (Berne: Peter Lang).

Franceschelli, S. AND Weil-Barais, A. (1999) Interactions professeur-élèves dans la construction d'un modèle en mécanique. In M. Gilly, J.P. Roux and A. Trognon (Eds.). *Apprendre dans l'interaction: analyse des médiations sémiotiques.* (pp. 241-257) (Nancy: Presses Universitaires de Nancy).

Galitz, W.O. (1997) *The essential guide to user interface design*. (NY: John Wiley & Sons, Inc.).

Gardner, M. (1970) 'The fantastic combinations of John Conway's new solitaire game of life*.' Scientific American*, Vol. 223, pp 120-123. (Mathematical Games, April 1970).

Gifford, B. and Enyedy, N. (1999) Activity Centered Design: Towards a Theoretical Framework for CSCL. In Proceedings of CSCL, 1999, pp. 189-197.

Gilbert, J.K. and Osborne, R.J. (1980) 'The use of models in science and science teaching.' European Journal of Science Education, *2*(1) pp:3-13.

Giordan, A. and De Vecchi G. (1987) *Les origines du savoir. Des conceptions des apprenants aux concepts scientifiques.* (Neuchâtel: Delachaux & Niestlé).

Goldman, S.V. (1992) *Computer Resources for Supporting Student Conversations about Science Concepts.* SIGCUE Outlook, 21 (3), p 4-7.

Greenman, J. (1994) 'Maps, Diagrams and Equations - the Stella approach to Mathematical Modelling'.

http://www.bham.ac.uk/ctimath/reviews/aug94/stella.pdf

Gutwin, C., Stark, G. and Greenberg, S. (1995) *Support for Workspace Awareness in Educational Groupware,* University of Calgary, Queen Elizabeth High School, Calgary, Alberta, Canada.

Harvard Educational Technology Center (1988) *Making sense of the future*. (Cambridge, Mass.: Harvard Graduate School of Education).

Harwood, D. (1995) The Pedagogy of the World Studies 8-13 Project: the influence of the presence/absence of the teacher upon primary children's collaborative group work. *British Educational Research Journal, 21* (5), pp 587-609.

Hoffheimer, M.H. (1982) Maupertuis and the Eighteenth-Century Critique of Preexistence. *Journal of the History of Biology*, 15, pp119-140.

Hoyles, C. (1995) Exploratory software, exploratory cultures? In A. A. diSessa, C. Hoyles and R. Noss. *Computers and exploratory learning*. (Berlin: Springer).

Jackson, S., Stratford, S., Krajcik, J., and Soloway, E. (1996b). Making dynamic modeling accessible to pre-college science students. *Interactive Learning Environments, 4*(3).

Jackson, S.L., Hu, J.T. and Soloway, E. (1994) ScienceWorks Modeler: Scaffolding the Doing of Science. In CHI'94 - Human Factors in Computing Systems, 1 (pp. 249-250). (Boston: ACM press).

Jackson, S.L., Stratford, S.J., Krajcik, J. and Soloway, E. (1995) Learner-Centered Software Design to Support Students Building Models. In American Educational Research Association Meeting. (San Francisco: USA).

Jackson, S.L., Stratford, S.J., Krajcik, J.S., and Soloway, E. (1996) Model-It: A case study of learner-centered design software for supporting model building. Technical Report: Center for Highly Interactive Computing in Education. Univ. of Michigan. Also at http://hi-ce-eecs.umich.edu/papers/

Jarboe, S. (1996) Procedures for enhancing group decision making. In B. Hirokawa and M. Poole (Eds), *Communication and Group Decision Making* (pp. 345-383). (Thousand oaks, CA:Sage Publications).

Jermann, P. (1999) *Structuring and regulating collaborative interaction by semi-structured interfaces and interaction meters,* TECFA, Faculty of Psychology and Educational Science University of Geneva, Switzerland.

Johnson-Laird, P.N. (1983). *Mental Models: Towards a cognitive science of language, inference and consciousness.* (Cambridge: Cambridge University Press).

Johnson-Laird, P.N. (1994) *L'ordinateur et l'esprit*. (Paris: Odile Jacob).

Jonassen, D. and Marra, R.M. (1994) Concept mapping and other formalisms as mindtools for representing knowledge. *ALT-J*, Vol. 2, pp.50-56.

Jonassen, D. and Reeves, T. (1995) Learning with Technology: Using computers as cognitive tools. *Handbook of Research for Educational Communications and Technology.* Macmillan Library.

Joshua S. and Dupin J.J. (1993) *Intoduction à la didactique des sciences et des mathématiques.* (Paris: Presses Universitaires de France) (Chapitre III : Les conceptions des élèves).

Katz, S. (1996) *Identifying the Support Needed in Computer-Supported Collaborative Learning Systems,* University of Pittsburgh.

Knowledge Revolution (1997) *Modellus, interactive modeling with mathematics*. (San Mateo, CA: Knowledge Revolution).

Komis V. and O'Flaherty, J. (2000) *Representation Tool Specifications*, Deliverable A06, Representation Consortium, Project Ref. MM 1045, June 2000.

Kreutzer, W. (1986) *Systems Simulation: Programming Styles and Languages.* (Wokingham, England: Addison-Wesley).

Kuipers, B. (1994). *Qualitative Reasoning: Modelling and Simulation with Incomplete Knowledge*. (Cambridge: The MIT Press).

Kumar, V.S. (1996) Computer supported collaborative learning: issues for research. http://www.uib.no/People/sinia/CSCL

Laurillard, D. (1995) Multimedia and the changing experience of the learner. *British Journal of Educational Technology 26*(3), pp 179-189.

Law, N. (1999) 'Making physics concepts accessible and explorable using WORLDMAKER an iconic modeling tool.' Workshop presented at ICPT99, Guilin, PRC.

Law, N. and Tam, C.W.E. (1998) 'WORLDMAKER(HK): An iconic modelling tool for children to explore complex behaviour'. International Conference on Computers in Education - ICCE98.

Law, N., Li, S., Li, R., and Tang, P. (2000) 'WORLDMAKER: Making collaborative exploration of complex systems accessible to children.', GCCCE 2000, Singapore.

Lawrence, I. and Whitehouse, M. (Eds.) (2000) *Advancing physics AS CD-ROM, teacher's version*. (Bristol, Institute of Physics).

Lawrence, I. and Whitehouse, M. (Eds.) (2001). *Advancing physics A2 CD-ROM, teacher's version*. (Bristol: Institute of Physics).

Lehtinen, E., Hallarainen, K., Lipponen, L., Rahikainen, M. and Muukkonen, H. (1998) *Computer Supported Collaborative Learning: A Review,* University of Turku, University of Helsinki.

Lemeignan, G. and Weil-Barais, A. (1987a) Apprentissage de la modélisation à propos de l'enseignement de la mécanique au lycée. In J.L. Martinand (sous la Dr de): *Enseignement et apprentissage de la modélisation: quantité de mouvement et modèle particulaire*. Rapport de fin de contrat: Action Concertée "Recherche en Education et en Formation".

Lemeignan, G. and Weil-Barais, A. (1987b). Construction inductive d'un modèle en mécanique et résolution de problèmes. In J. Colomb and J.F. Richard (Eds): *Résolution de problèmes en mathématiques et physique* (pp. 145-198), Collection Rapport de Recherches, n° 12. (Paris: I.N.R.P.).

Lemeignan, G. and Weil-Barais, A. (1989) Enseignement et apprentissage d'un concept par les élèves: La quantité de mouvement en classe de seconde. *Bulletin de l'Union des Physiciens, 716*, pp1013-1030.

Lemeignan, G. and Weil-Barais, A. (1990) *L'apprentissage de la modélisation dans l'enseignement de l'énergie*. Rapport de fin de contrat LIREST- INRP, Enseignement et apprentissage de la modélisation. 90 pages et 19 pages d'annexes.

Lemeignan, G. and Weil-Barais, A. (1992) L'apprentissage de la modélisation dans l'enseignement de l'énergie. In Equipe LIREST/INRP: *Enseignement et apprentissage de la modélisation en sciences*. (Paris: INRP)

Lemeignan, G. and Weil-Barais, A. (1993) *Construire des concepts en physique; l'enseignement de la mécanique.* (Paris: Hachette).

Lemeignan, G. and Weil-Barais, A. (1994). Developmental approach to cognitive change in mechanics. *International Journal of Science Education, 16*, 1, pp 99-120.

Lewis R. (1998) 'Learning together: a rationale, some experiences, and a framework' in Rouet, J.F. and B. de la Passardière (Eds) *Hypermédias et Apprentissages*.

Mandinach, E.B. (1989) 'Model-Building and the Use of Computer Simulation of Dynamic Systems'. *Journal of Educational Computing Research*, *5*(2), pp.221-243.

Maragoudaki, E. and Boohan, R. (1999) 'Models, rules and behaviours: an investigation of the way that children approach a modelling program', in Koumaras, P, Kariotoglou, P, Tselfes, V and Psillos D (eds) Proceedings of the First Panhellenic Conference on 'Science Education and Applications of New Technologies in Education', Thessaloniki, Greece, May 29-31, 1998, (Thessaloniki: Christodoulides Publications), pp 338-344 ISBN 960 7577 24 8

Maragoudaki, E., Boohan, R. and Ogborn, J. (1997) 'Computer models, rules and behaviours: children's understanding of structures' Paper presented at EARLI Conference, Athens, August 1997.

Martinand, J.L., Chomat, A., Larcher, C., Lemeignan, G., Meheut, M., Weil-Barais, A. (1991). Développement d'activités de modélisation dans l'enseignement des sciences physiques- Lycées et Collèges. Rapport de fin de contrat, DESUP, 38 pages. LIREST.

Marx, G. (1981a) 'Some Simulations of Science I'. *Physics Education*, Vol. 16, pp. 152-158.

Marx, G. (1981b) 'Some Simulations of Science II'. *Physics Education*, Vol. 16, pp. 212-217.

Marx, G. (1984a) Simulation Games in Science Education. *European Journal of Science Education*, Vol. 6, 31-45

Marx, G. (1984b) *Games Nature Plays*. Roland Eötvos University, Budapest.

Mathematica web site, http://www.wolfram.com, (Wolfram Research Inc).

Matlab web site, http://www.mathworks.com/, (Mathworks Inc).

McAleese, R. (1998) The Knowledge Arena as an Extension to the Concept Map: Reflection in Action. *Interactive Learning Environments*, Vol. 6, No 1, pp. 1-22.

McManus, M.M. and Aiken, M.R. (1995) Teaching collaborative skills with a group leader computer tutor, *Education and Information Technologies* Vol. 1, pp.75-96 (1996).

Mellar, H. and Bliss, J. (1994) Introduction: Modelling and Education. In H. Mellar, J. Bliss, R. Boohan, J. Ogborn, and C. Tompsett (Eds.), *Learning with artificial worlds: computer based modeling in the curriculum.* (London: The Falmer Press). pp 1-7.

Mellar, H., Bliss, J., Boohan, R., Ogborn, J. and Tompsett C. (Eds.), *Learning with artificial worlds: computer based modeling in the curriculum.* (London: The Falmer Press).

Microsoft (1995) *The Windows interface guidelines for software design*. (Redmond: Microsoft Press).

Miller, P. (1983). *Theories of developmental psychology* (3rd Edition). (New York: Freeman).

Miller, R., and Ogborn, J. (1993). Educational Tools for Computational Modelling. *Computers and Education*, *21*(3), pp 205-261.

Miller, R.S., Ogborn, J., Turner, J., Briggs, J.H. and Brough, D.R. (1990). 'Towards a Tool to Support Semi-Quantitative Modelling'. In International Conference on Advanced Research on Computers in Education, Tokio, Japan.

Minsky, L.M. (1968) 'Matter, Mind, and Models', in Minsky, L.M. *Semantic information processing*, 1st ed. (MIT Press).

Moeschler, J. (1985) *Argumentation et Conversation,* (Paris: Credif-Hatier).

Moore, D. (1993) Theory of transactional distance. In D. Keegan (Ed), *Theoretical principles of distance education* (pp. 22-38). (London: Routledge).

Nakamura, M., Hanamoto, K., and Otsuki, S. (1999). Assistance and Visualization of Discussion for Group Learning, *Artificial Intelligence in Education,* S.P. Lajoie and M. Vivet (Eds.) (IOS Press, 1999).

Nardi, B. (1996) (Ed.) *Context and Consciousness, Activity Theory and Human-Computer Interaction.* (The MIT Press).

NCTM (1989) *Curriculum and evaluation standards for school mathematics.* (Reston: VA, USA)

NCTM (2000) *Curriculum and evaluation standards for school mathematics.* (Reston: VA, USA)

Nemirovsky, R. and Rubin, A. (1992a) *Students' tendency to assume resemblances between a function and its derivative*, Working Paper 2-92. TERC, Massachusetts Ave, Cambridge, Massachusetts.

Nemirovsky, R. and Rubin, A. (1992b) 'It makes sense if you think about how the graphs work, but in reality...', Internal Working Paper. TERC, Massachusetts Ave, Cambridge, Massachusetts

Neumann, Von J. (1947) 'The Mathematician' in R.B. Heywood (ed) *The Works of Mind*, 1st ed. (The University of Chicago Press).

Norman, D.A. (1993) *Things that make us SMART*. (Reading, MA.: Addison-Wesley).

Novak, J. (1977) *A theory of education.* (Cornell University Press).

Novak, J. (1990) Concept maps and Vee diagrams: two metacognitive tools to facilitate meaningful learnin. *Instructional Science*, Vol. 19, pp. 29-52.

Novak, J. (1998) *Learning, Creating and using knowledge: Concept maps as facilitative tools in schools and corporations*. (Lawrence Erlbaum).

Novak, J. amd Gowin, D.B. (1984) *Learning how to learn*. (Cambridge University Press).

Nwana, H.S. (1996) Software Agents: An Overview. *Knowledge Engineering Review*. Vol.11, No3, pp.205-244.

Ogata, H., Matsuura, K. and Yano, Y. (2000) Active knowledge Awareness Map: Visualizing learners activities in a Webb based CSCL environment, *International Workshop on New Technologies in Collaborative Learning,* Tokushima, Japan.

Ogborn, J. (1990) A Future for Modelling in Science Education. *Journal of Computer Assisted Learning*, *6*, pp 103-112.

Ogborn, J. (1994) The design of exploratory and expressive learning environments. In R. Lewis and P. Mendelsohn *Lessons from learning*. (Amsterdam: North-Holland) pp 125-135.

Ogborn, J. (1999) Modeling clay for thinking and learning. In W. Feurzeig and N. Roberts (Eds) *Modeling and Simulation in Science and Mathematics Education* (New York: Springer-Verlag) pp 5-37.

Ogborn, J. and Miller, R. (1994) 'Computational issues in modelling.' In H. Mellar, J. Bliss, R. Boohan, J. Ogborn, and C. Tompsett (Eds.), *Learning with artificial worlds: computer based modeling in the curriculum.* (London: The Falmer Press). pp 33-38.

Ogborn, J., and Whitehouse, M. (Eds.) (2000) *Advancing physics AS*. (Bristol: Institute of Physics).

Ogborn, J., and Whitehouse, M. (Eds.) (2001) *Advancing physics A2*. (Bristol: Institute of Physics).

Olguin, C.J.M., Delgado A.L.N. and Ricarte I.L.M. (2000) An Agent Infrastructure to set Collaborative Environments. *Educational Technology & Society*, *3*(3), pp. 65-73.

Osborne, R.J., and Gilbert, J.K. (1980) The use of models in science teaching. *The School and Science Review, 62*, pp: 57-67.

Papert, S. (1980) *Mindstorms, Children, Computers and Powerful Ideas*. (Basic Books).

Pea, R.D. (1985) 'Beyond amplification: Using the computer to reorganize mental functioning', *Educational Psychologist*, Vol:20, No:4, pp:167-182.

Pea, R.D., Edelson, D. and Gomez, L., (1995). *The CoVis Collaboratory: High school science learning supported by a broadband educational network with scientific visualization, videoconferencing, and collaborative computing,* Northwestern University.

Perkins, D. (1993) Teaching for understanding. *American Educator 17*(3), pp: 8, 28-35.

Piaget, J. (1946) *La formation du symbole chez l'enfant*. (Neuchâtel: Delachaux & Niestlé).

Piaget, J. (1974) *La prise de conscience*. (Paris: PUF).

Piaget, J. and Inhelder, B. (1955) *De la logique de l'enfant à la logique de l'adolescent*. (Paris: PUF).

Pilkington, R. (2001) Analyzing Educational Dialogue Interaction: Towards Models that Support Learning, *International Journal of Artificial Intelligence in Education, (20001), 12*, 1-7.

Ploetzner, R., Spada, H., Stumpf M. and Opwis, K. (1990) 'Learning qualitative and quantitative reasoning in a microworld for elastic impacts.' *European Journal of Psychology of Education*, Special Issue: 'Psychology of Learning Physics', 5, pp 501-516.

Politis, P., Komis, V., Dimitracopoulou, A. (2001) MODELSCREATOR un logiciel de modélisation permettant l'utilisation des règles logiques et la prise de décision. E.P.I., Juin.

PowerSim web site, http://www.powersim.com/

Quillian, M.R. (1968) Semantic Memory, in Minsky M. (ed.), *Semantic Information Processing*. (MIT Press) pp. 459-475.

Rahkila, M. and Karjalainen, M. (1999) *Evaluation of learning in computer based education using log files,* 29th ASEE/IEEE Frontiers in Education Conference, November 1999.

Repenning, A. (2001) 'Programming by Analogous Examples'. In Lieberman H. (Ed.) *Your Wish Is My Command: Programming by Example*. Vol. 43, pp. 90-97. (Morgan Kaufmann Publishers).

Repenning, A. and Summer, T. (1994) 'Programming as Problem Solving: A participatory theater approach'. Workshop on advanced visual interfaces '94, Bary, Italy, pp. 182-191.

Repenning, A., Ioannidou, A. and Phillips, J. (1999) 'Collaborative Use and Design of Interactive Simulations'. Proceedings of Computer Supported Collaborative Learning Conference at Stanford (CSCL'99).

Repenning, A., Ioannidou, A. and Zola, J. (2000) 'AgentSheets: End-User Programmable Simulations'. *Journal of Artificial Societies and Social Simulation*, 3(3).

Representation Project (2000) *Deliverable A12*. Representation Consortium, Project Ref. MM 1045, July 2000

Resnick, M. (1991) 'Animal simulations with *Logo: Massive parallelism for the masses'. In J.A. Meyer, and S. Wilson (Eds.) *From animals to animats*. (Cambridge, MA: MIT Press).

Resnick, M. (1992) 'Beyond the centralized mindset: Explorations in massively parallel microworlds'. Unpublished PhD dissertation. MIT.

Rich, E. and Knight, K. (1991) *Artificial Intelligence*, 2nd ed., (McGraw-Hill).

Richmond, B. *et al.* (1985) *STELLA. [Computer Program].* (Lyme: High Performance System, Inc.).

Richmond, B. *et al.* (1987) *An Academic User's Guide to STELLA.* (Lyme: High Performance System, Inc.).

Richmond, B. *et al.* (1997) *STELLA Introduction to Systems Thinking.* (Lyme: High Performance System, Inc.).

Roberts, N. *et al.* (1983) *Introduction to Computer Simulation - A System Dynamics Modelling Approach.* (New York: Addison Wesley)

Robertson, J., Good, J. and Pain, H. (1998) BetterBlether: The Design and Evaluation of a Discussion Tool for Education, *International Journal of Artificial Intelligence in Education, (1998),* 9, 219-236.

Robinson, A.J. (1965) 'A machine-oriented logic based on the resolution principle', JACM, Vol:12, pp:23-41.

Roschelle, J., DiGiano, C., Koutlis, M. and Repenning, A. (1999) 'Developing Educational Software Components'. *IEEE Computer*, Vol.32, pp.50-58.

Rumelhart, D.E. and McLelland, J.l. (1987) *Parallel Distributed Processing - Explorations in the Microstructure of Cognition. Volume 1: Foundations* (3rd. ed.). (Cambridge, MA: MIT press).

Salomon, G. and Globerson, T. (1989) When teams do not function the way they ought to, *International Journal of Educational Research,* 13, 89-98.

Sampaio, F. Ferrentini (1996) LinkIt: design, development and testing of a semi-quantitative computer modelling tool. Unpublished Ph.D. Thesis, University of London.

Sampaio, F. Ferrentini and Ogborn, J. (1996). Linkit: A Modelling Tool without Mathematics. *The Thirteenth International Conference on Technology and Education. Proceedings* Volume 1. March 17 – 20, 1996. New Orleans, Louisiana.

Sampaio, F. Ferrentini, Santos, A.C. Kurtz dos and Ferracioli, L. (1999) Semi-Quantitative Computer Modelling and Classroom Science. In: International Conference On Mathematics/Science Education & Technology, Charlottesville, VA. *Proceedings of M/SET99*. (San Antonio, Texas: AACE - Association for the Advancement of Computing in Education) pp513-518.

Santos, A.C. Kurtz dos (1992) Computational Modelling in Science Education: A Study of Student's Ability to Manage Some Different Approaches to Modelling. Unpublished Ph.D. Thesis, University of London.

Scardamalia, M. and Bereiter, C. (1989) *Schools as knowledge-building communities*, Paper presented at the Workshop on Development and Learning Environments, University of Tel Aviv, Tel Aviv, Israel, October, 1989.

Scardamalia, M. and Bereiter, C. (1994) Computer Support for Knowledge-Building Communities. *The Journal of Learning Sciences*, *3*(3), pp. 265-283.

Schwartz, J. L. (1989) Intelectual mirrors: A step in the direction of making schools knowledge-making places. *Harvard Educational Review 59*(1), pp 217-222.

Schwartz, J.L. (1993) A Personal View of the Supposer: Reflections on Particularities and Generalities in Educational Reform. In J.L. Schwartz, M. Yerushalmy and B. Wilson *The geometric supposer: what is the case of?.* (Hillsdale, NJ: Erlbaum).

Shum, P. (1997) The Effect of engaging in iconic modelling on the learning of science. Unpublished MEd dissertation. University of Hong Kong.

Simoff, S. (1999). Monitoring and evaluation in Collaborative Learning Environments, *Proceedings of the AAAI Fall Symposium: Learning How to Do Things,* Cape Cod, MA.

Slavin, R.E. (1990) Research on cooperative learning: Consensus and controversy, *Educational Leadership, 47 (December 1989/January 1990),* 52-54.

Smyrnaiou, Z. (2000) *Expérimentation et Conceptualisation*. DEA, Université Paris V.

Soller, A. and Lesgold, A. (2000) Knowledge acquisition for adaptive collaborative learning environments, *Proceedings of the AAAI Fall Symposium: Learning How to Do Things,* Cape Cod, MA.

Soller, L.A. (1999) *Supporting Social Interaction in a Intelligent Collaborative Learning System,* University of Pittsburgh.

Soloway, E., Pryor, A.Z., Krajcik, J.S., Jackson, S., Stratford, S.J., Wisnudel, M. & Klein, J.T. (1997) 'ScienceWare's Model-It: Technology to Support Authentic Science Inquiry'. *T.H.E. Journal*, Vol. 25, No. 3, pp. 54-56.

Spitulink, M.W., Krajcik, J. and Soloway, E. (1999) Construction of models to promote scientific understanding. In W. Feurzeig and N. Roberts (Eds) *Modeling and Simulation in Science and Mathematics Education* (New York: Springer-Verlag) pp70-94.

Spitulnik, M.W. (1998) Construction of technological artifacts and teaching strategies to promote flexible scientific understanding. Unpublished Doctoral Dissertation, University of Michigan.

STARLOGO web site, http://www.media.mit.edu/starlogo/

Steed, M. (1992) 'STELLA, A Simulation Construction Kit: Cognitive Process and Educational Implications'. *Journal of Computers in Mathematics and Science Teaching*, 11, pp 39-52.

Stern, J. (2000) The Design of Learning Software: Principles Learned from the Computer as Learning Partner Project. *Journal of Science Education Technology*, Vol. 9, No1, pp. 49-65.

Stevens, T.R. (1990) *Fractal programming in Turbo Pascal*, (California M&T Publishing Inc).

Stratford, S. (1996). Investigating processes and products of secondary science students using dynamic modeling software. Unpublished Doctoral Dissertation, University of Michigan.

Suthers, D. (1998). *Representations for Scaffolding Collaborative Inquiry on Ill-Structured Problems,* University of Pittsburgh.

Suthers, D. (1999) *Representational Support for Collaborative Inquiry*, Proceedings of the Hawai' in International Conference on System Sciences, IEEE, 1999.

Suthers, D. and Jones, D. (1997) An architecture for intelligent collaborative educational systems. *AI-Ed-97, the 8th World Conference in Artificial Intelligence in Education,* Kobe Japan, August 20-22, 1997.

Suthers, D.D., Toth, E.E. and Weiner, A. (1997) An Integrated Approach to Implementing Collaborative Inquiry in the Classroom, *Computer Supported Collaborative Learning '97,* Dec. 10-14, 1997.

Teodoro, V.D. (1994) 'Learning with Computer-Based Exploratory Environments in Science and Mathematics'. In S. Vosniadou, E. de Corte, and H. Mandl (Eds.), *Technology-Based Environments: Psychological and Educational Foundations*, NATO ASI Series, F: Computer and Systems Sciences 137. (Berlin: Springer Verlag) pp 179-186.

Teodoro, V.D., Vieira, J.P.D. and Clérigo, F.C. (1997) *Modellus, interactive modelling with mathematics [Software for Windows]*. San Mateo, (CA: Knowledge Revolution).

Teodoro, V.D., Vieira, J.P.D., and Clérigo, F.C. (2000). *Modellus 2.01: Interactive modelling with mathematics [Software for Windows]*. (Monte de Caparica: College of Science and Technology, New University of Lisbon).

Thornton, S. (1997) 'Karl Popper' *Stanford Encyclopedia of Philosophy*, Stanford University.

Toffoli, T. and Margolis, N. (1987) *Cellular automata machines: A new environment for modelling.* (Cambridge, Massachussets: MIT Press).

Tompsett, C. (1994) 'An Introduction to qualitative modelling' In H. Mellar, J. Bliss, R. Boohan, J. Ogborn, and C. Tompsett (Eds.), *Learning with artificial worlds: computer based modeling in the curriculum.* (London: The Falmer Press). pp 145-152.

Veerman, A. (2000) *Computer Supported Collaborative Learning Through Argumentation,* Proefschrift Utrecht University, ISBN: 90-393-2393-3.

Veerman, A.L. and Treasure-Jones, T. (1999) Software for problem-solving through collaborative argumentation. In P. Coitier and J.E.B. Andriessen (Eds), *Foundations of* *argumentative text processing* (pp. 203-230). (Amsterdam: Amsterdam University press).

Vergnaud, G. (1987) La fonction de l'action et de la symbolisation dans la formation des connaissances chez l'enfant. In J. Piaget, P. Mounoud & J.P. Bronckart (Eds), *Psychologie*, 821-844, Encyclopédie de la Pléiade. (Paris: Gallimard).

Vergnaud, G. (1990) La théorie des champs conceptuels. In *Recherches en Didactique des Mathématiques*, *10*, 2-3, 133-170.

Vergnaud, G. (1994) *Apprentissages et Didactiques, où en est-on*? (Paris: Hachette).

Vergnaud, G., Halbwachs, F. and Rouchier, A. (1978) Structure de la matière enseignée, histoire des sciences et développement conceptuel chez les élèves, *Revue Française de Pédagogie, 45*, 7-15.

Vygotski, L. (1985a) *Pensée et langage*. Trad. Fr. Sève. (Paris: Messidor/Éditions Sociales).

Vygotski, L. (1985b) *La pensée et le mot*. In B. Schneuwly et J.P. Bronckart (Eds.) *Vygotski aujourd'hui.* (Neuchâtel: Delachaux et Niestlé)

Webb, N.M. (1987) Peer interaction and learning with computers in small groups, *Computers in Human Behavior, 3*, 193-209.

Weil-Barais, A. (1991) *Développement d'activités de modélisation dans l'enseignement des sciences physiques- Lycées et Collèges*. LIREST.

Weil-Barais, A. (1993) *L'homme cognitif*. (Paris: PUF).

Weil-Barais, A. (1994a) Heuristic value of the notion of zone of proximal development in the study of child and adolescent construction of concepts in physics. *European Journal of Psychology of Education, 9*(4)*,* pp 367-383.

Weil-Barais, A. (1994b) *Étude de l'impact de l'utilisation d'outils informatiques par les élèves en sciences physiques*. Rapport de fin de contrat: Convention d'étude et de recherche avec le Ministère de l'Éducation Nationale, Direction des Lycées et des Collèges, Direction de l'Informatique et des Technologies Nouvelles.

Weil-Barais, A. (1997) De la recherche sur la modélisation à la formation des professeurs de physique : comment s'opère la transition ? *Shkolé, Cahiers de la recherche et du développement, 7,* pp 141-154.

Weil-Barais, A and Lemeignan, G. (1989) A propos de recherches sur l'enseignement et l'apprentissage de la modélisation en sciences physiques. *Cahiers de Beaulieu Recherches et informations pour l'enseignement des disciplines scientifiques et techniques, 9*, pp 33-47.

Weil-Barais, A. and Lemeignan, G. (1990) Apprentissage de concepts et modélisation. *European Journal of Psychology of Education, 5*, pp 391-415.

Weil-Barais, A. and Lemeignan, G. (1994) Approche développementale de l'enseignement et de l'apprentissage de la modélisation; quelles implications pour la recherche et pour les pratiques d'enseignement? In Equipe INRP/LIREST: *Nouveaux regards sur l'enseignement et l'apprentissage de la modélisation en sciences.* (pp. 85-113) (Paris: INRP).

Whitfield, A.H. (1988) 'STELLA and its Impact on the Teaching of Mathematical Modelling'. *Computers in Education*, pp 299-304.

Winograd, T. (Ed.) (1996) *Bringing design to software*. (Reading, Mass.: Addison-Wesley).

Wolfram, S. (1984) 'Cellular automata as models of complexity'. In *Nature* (4th Oct 1984).

WORLDMAKER(HK) web site: http://worldmaker.cite.hku.hk/worldmaker/pages/

Zaraza R. and Fisher D. (1996). Experiences in Developing Single-Discipline and Cross-Curricular Models for Classroom Use, International System Dynamics Conference, Cambridge, MA

Zaraza R. and Fisher D. (1997) Introducing system dynamics into the traditional secondary curriculum: The CC-STADUS project's search for leverage points. International System Dynamics Conference, Istanbul, Turkey.

PART III

REQUIREMENTS AND CONSIDERATIONS FOR A NEW MODELLING ENVIRONMENT

This chapter presents and argues the central requirements and considerations related to modelling activities and collaboration in real school settings. For this purpose it takes into consideration the analysis of the previous chapter and particularly teachers' general software requirements, national educational policies and curricula, theoretical framework based on research results, the review of existing systems, etc. Additionally, it takes into account and makes references to three research results that have been carried out in the frame of ModellingSpace project, in order to explore specific points. It provides the main considerations for the determination of design principles and initial specifications that will be presented in Part IV of this document.

In its first sub-part, it explicates the main reasons for conceiving and designing a new modelling environment, and therefore the innovation of ModellingSpace provided activities. It specifies requirements related to inquiry learning and modelling support. It points out the importance of self-regulation, scaffolding and metacognitive support. Finally it argues about the teachers' roles and needs during modelling activities in school settings.

The second sub-part is devoted to the ModellingSpace's envisaged support to collaborative learning and to the learning community. It firstly presents explicitly the reasons why ModellingSpace project focuses on collaborative learning and the creation of a learning community. Then, it highlights the importance of appropriate means of action, coordination and awareness, as well as the importance of appropriate means of dialogue during synchronous and/or asynchronous collaborative learning activities. It focuses on the role and needs of teachers in order to manage these activities as well as on the necessity to provide specific analysis tools for teachers. Finally, it presents our vision of the emerging technology-based learning community and of how to support it.

1. Requirements related to technology-based modeling activities

1.1 Main reasons to design a new modelling environment

There are four main reasons that have led our consortium to propose and conceive a new technology-based learning environment that promotes modeling activities.

1. **Epistemological reasons**

Scientific activity involves to a great extend creation, validation and application of appropriate models of the phenomena, systems or situations under study. These models can be either formal (such as the mathematical models built in physics) or can constitute simple iconic representations.

Models appear in most scientific areas (economics, history, biology, meteorology, archaeology etc.), as well as in our everyday life.

Moreover, during the last years, modelling tools, provided by computer science, have considerably influenced the work of some disciplines. Computers have amplified the power of traditional models, but have also provided new representation systems and conceptual **frameworks** for modelling. Efficient employment and management of modelling tools appear to be key capabilities for the future citizen

(For more, see current document, Part II, Chapter 1: Models and modelling in science.)

2. Learning order reasons

Teachers and students of all levels often conceive science education as a process of information transfer in which students accumulate passively whatever fact is conveyed to them by an instructor or by a text book. This process is inherently unproductive, and its purpose is further defeated by the wrong assumptions that students are originally motivated and equally ready to learn transferred knowledge meaningfully.

Educational research has been constantly showing that a good proportion of high school and college students experiences the following drawbacks (Halloun, 2000):

- Wrong views about the nature of science in general, and its relation with the real world and every day life.
- Tenacious, situation-specific, vague, mutually confused, fragmented, loose and incoherent ideas about for instance physical systems and phenomena/
- Learning by rote with a concentration on the memorization of mathematical formulas and routines and through a blind imitation of authority, be it a teacher or the textbook.
- Formula-based, trial and error problem-solving tactics.

Furthermore, when we present directly to students already conceived models (e.g. algebraic ones), they don't appreciate the value of the model, they are not able to appreciate the significance of variables, and thus they often use models outside the conditions of their validity, or cannot evaluate the appropriateness of the results of the application of a model.

During the last decade, research in the field of science education and cognitive psychology [among others Clement 1989, Martinand, 1992; Lemeignan and Weil-Barais, 1993; Bliss, 1994; Halloun, 1996] has indicated that the application of a modelling process could reinforce the learning process for a number of reasons:

- Through a model construction process, learners express their own ideas and mental models [Bliss, 1994] of which, in most cases, they are not aware. This expression is the first step towards the process of cognitive awareness of ideas and reasoning modes, which are often necessary for conceptual change [Vosniadou, De Corte and Mandl, 1994].
- The graphical and iconic representations that the models can obtain enable the abstract ideas to acquire a concrete form. These representations play the role of thinking support, a role that accompanies thought and reasoning [Laborde and Vergnaud, 1994; Teodoro 1997].
- The expression of thoughts through model construction can help the learning process, since the ideas become an object of communication and discussion.

(For more, see current document, Part II, Chapter 2, Sections 2.1-2.2: Models and modelling as a cognitive activity.)

3. Reasons related to the recent curricula and educational policies in various countries

- During the last few years, national curricula of various European countries, have recognised the importance of modelling activities, specially in relation to science, mathematics, and technology education.
- At the same time, more and more curricula incorporate explicitly, and in a relatively systematic way scopes and activities related to information and communication technologies. Moreover, national curricula explicitly define the goal that "Pupils should be able to use information technology to design, develop, explore and evaluate models of real and imaginary situations". (http://www.nc.uk.net/home.html)

- Additionally, recent curricula incorporate interdisciplinary teaching approaches for the subjects studied in schools. The use of models and modelling processes constitutes a common point among different disciplines. One of the worldwide problems of the current curriculum is the fragmentation of knowledge among different subject areas. Modelling activities could contribute to the unification of common points between different subject areas, and could promote interdisciplinary teaching approaches.

(For more, see current document, Part II, section 2.4: Models and modelling in European curricula.)

The above depict modelling activities as an ongoing important dimension in the school curriculum and suggest the design of appropriate technology-based learning environments.

4. Limitations of existing modelling environments

- Modelling activities could contribute to the unification of common points between different subject areas, and could promote interdisciplinary teaching approaches. Consequently, we are interested in a combined environment that offers different kinds of models which could support a wide range of activities in different subject areas. Existing well-designed systems are focused in specific domain areas (e.g. the learner-centred modelling environment Model-IT, is focused mainly on ecology).
- An appropriate modelling environment should allow pupils to externalise and act on their thoughts and ideas, by providing appropriate and flexible structures. Nevertheless, in order to support these kinds of reasoning it is essential to create environments appropriate for young children that will not be based on abstract modelling and representation tools. (E.g. Well-designed and powerful modelling environments such as Modellus, or even SIMQUEST do not seem appropriate for young students of primary school).
- It is significant to emphasise qualitative and semi-quantitative reasoning, since research in the field of cognitive psychology has proved their importance in the learning process and knowledge development [Bliss 1994; Soloway *et al*., 1994]. For this purpose, qualitative, semi-quantitative and quantitative models should be supported by the same environment. Furthermore, the whole environment must really support the transition from qualitative or semi-quantitative reasoning to quantitative one (e.g. ModelsCreator, designed to allow all three kinds of modelling, does not fully support this transition).
- It is really worthwhile to take advantage of the new and powerful learning possibilities that the current new technologies of communication offer (existing systems such as, Modellus, Model-IT, and ModelsCreator, seem actually limited, given that they do not support a wider learning community, flexibility of modes and conditions of use, and collaborative learning).

(For more, see current document, Part II, Chapter 4, Analysis of modelling environments.)

Having accepted that there is an educational interest in designing and producing educational software for modelling, we need to give answers to some central questions which concern the design, development and appropriate pedagogical approaches of use:

- What are the design requirements concerning human computer interaction in order to produce a modelling system appropriate for young students?
- Which type of reasoning modes and what kind of models should be promoted?
- By surpassing the drawbacks of existing modelling systems, how can modelling tools be conceived in order to be appropriate for young students?
- What are the needs of classroom management that could be taken into account in the system design?

- Concerning the use of modelling software in the current school settings, how can we cope with a new learning environment and the associated new pedagogical and didactical strategies? How could we support teachers in these aspects?
- How could we benefit from the collaborative learning possibilities, as well as from the motivational and cognitive aspects that a wider learning community could offer?

1.2 Modelling and Reasoning: categories of modelling formalisms and kinds of relations adopted

Scientists, engineers and technicians are frequently called upon to apply their expertise to new domains of knowledge. What is needed in order to foster such transfer is: a) an understanding of forms of models that are applicable in multiple domains b) inquiry skills for developing models and evaluating their appropriateness within a domain and c) generic reasoning strategies used in applying models when solving problems.

In order to apply appropriate modelling to different problem categories and scientific fields, different modelling formalisms have been developed (Ogborn 1994): difference equations, algebraic structures, finite elements, statistical models, geometric models, graph theory, Monte Carlo methods, cell automata, production systems, discrete event models, logical formalisms, etc.

But among the different modelling formalisms what are these that appear the most appropriate in order to be used by young students for the modeling of a wide range of phenomena and problems and in different subject matters of the school curricula?

Four main categories of modelling formalisms are perceived by ModellingSpace as the most appropriate for young children in order to work in the frame of their existing school curricula:

A) *Quantitative models or mathematical algebraic modelling formalism*: This formalism is used in various disciplines. Specifically, in mathematics and in sciences quantitative models are central throughout senior high school. Quantitative models make use of quantifiable variables and algebraic relations. In all quantitative models the initial conditions are specified by giving values to independent variables. The model uses algebraic relationships to calculate the values of depended variables. The behaviour of the model depends on both the values of the independent variables and on the nature of relationships between the variables.

B) *Semi-quantitative modelling formalisms:* Semi-quantitative models involve quantifiable variables, whose change however is not defined by algebraic relationships, but by the kind of influence that one exerts on the other. In other words these models are based on a formalism that indicates qualitative relationships. Expressions of semi-quantitative reasoning will be supported by MODELLINGSPACE in order to allow children to express themselves in an easier way and to help them gradually achieve the creation of quantitative models, as well as in order to allow them to analyse the relations and influences of factors intervening in complex phenomena.

   As the requirements of teachers working on ModelsCreator indicate (see, current document, Part II, section 6.4), there is a need to distinguish and support two different kinds of semi-quantitative relations:

   i) *Semi-quantitative formalisms and relations that correspond to known and specific types of algebraic relations* (e.g. to y=a*x), This kind of formalism is appropriate to phenomena and problems studied in school physics.

   ii) *Semi-quantitative formalisms and relations that do not correspond to any pre-defined algebraic relation,* but indicate only the way that one variable may influence another one. This formalism is appropriate to model phenomena and problems studied by environmental education and biology.

C) *Semantic qualitative modelling formalisms constituting concept maps*: They form static, non executable models, like concept maps. Qualitative models express relationships which cannot be expressed in a quantifiable way, and of which the criteria of validity are not strictly defined. Such relationships appear in all the subject matters of school curricula. For instance, the creation of a concept map to present the concepts (and their relations) of a specific domain, is always a valuable learning activity either for purposes of diagnosis of alternative conceptions (e.g. at the beginning of a unit) or of synthesis of acquired concepts (e.g. at the end of a unit).

Additionally, there are situations in which using concept map-like diagrams for a qualitative analysis of the implicated factors is very important. For instance, scientists have proposed that the production of a forces diagram in mechanics (physics) should be based on a qualitative analysis of the system's objects and their interactions, represented in a diagrammatic form similar to this of a concept map (Dumas-Carré and Caillot, 1989, Dimitracopoulou and Dumas-Carré, 1996). Similarly, diagrammatic analyses of energy transfer chains have also been proposed and used as cognitive aids (Weil-Barais and Lemeignan, 1990). Moreover, their use has been extensively studied by researchers in both laboratory and technology based collaborative learning environments (Tiberghien, 1996; Baker *et al.*, 1999).

However, it is to be noted that the most typical and extended use of qualitative analysis expressed by concepts maps appears in environmental education, where more complex systems without known mathematical equations are studied, (e.g. in order to find the factors affecting pollution).

D) *Qualitative decision making based on logical operators*: These relations and their formalism is based on logical operators (AND, OR, NOT) and control conditions (if…then), useful in decision making. In decision making, the qualitative reasoning through logical operators, appears important in subject matters such as Economy, Professional Orientation and Computer Sciences. It provides a formalism that could help students in structuring and ordering ideas and relationships and could enable modelling of decision making (Huitt, 1992). The system must offer executable models based on event-based simulations where the events depend on decisions (made by the user), rules, or even probabilities.

Detailed analysis and presentation of this kind of models and associated reasoning modes, as these might correspond to different ages and subject matters will be reported in the deliverable 'Scenarios and activities' (DO5).

1.3 How to support students' expression and reasoning: considerations related to entities and variables implicated in models

A) How to encourage students to express their ideas and proceed by their own conceptualisation of the situation under study? *By offering them a wide range of 'variables', and not just the scientific ones, to express their ideas with.*

The latest approaches to learning suggest that we must render children able to express their intuitive ideas and test their validity in order to change and/or gradually develop them. Consequently, the technological learning environment must allow children to express these ideas. However, children's ideas are not always in accordance with the scientific ones, nor are they a subset of them.

For scientists, the initial analysis and description phase in a problem solving or modelling process is severely constraint by their choice of theory to be applied (for instance, mechanics); this specifies what kind of objects and properties can be modelled. The main outputs of the descriptive phase are a complete set of names and descriptive variables for the model, along with physical interpretations and correspondences for all the variables. For instance, in a description of motion the state variables of the model are specified, such as position and velocity. In a description of an interaction each agent acting on the object considered as the system is identified, along with the type of interaction. Thus, what is needed to be decided is how to model the objects (for instance as material points), and to infer the concepts which would permit an appropriate and complete description of the phenomenon in relation to the initial questions (what variables to use in order to represent the properties that influence the phenomenon under study).

However, reality can be viewed without any kind of 'scientific' concepts. Students, and specially those who are in the process of constructing scientific concepts, can interpret reality, simply as constituted by objects (such as inclined plan, ball, person) (Chi, Feltovich and Glaser, 1981). Most people's everyday thinking is about real entities and events (Bliss, 1992), even if scientific models are more abstract than reality, and a lot has to be left out. Young students' thinking is 'concrete object oriented' and not 'concept oriented'.

When a simulation-based or modelling software is designed, the designers should make all that is needed in order to provide the possibilities to create and test a number of the known scientific models, using scientific terms and symbols. Most modelling or simulation systems impose directly abstract thinking and particularly the use of variables. Moreover, in order to allow students to explore a phenomenon by manipulating the relevant factors, they present them directly with the whole list of the implicated variables (not less not more). For instance, they present in an explicit way in the menu the variables of time, distance, velocity, mass, etc.. This situation reduces the possibility that students will reflect on their own cognitive resources.

It is important to keep away from the eventually technical restrictions, and thus in the question 'How to encourage students to express their ideas and proceed by their own conceptualisation of the situation under study?' we have to answer: By offering them a wide range of 'variables', and not just the scientific ones, to express their ideas with.

Concerning the entities MODELLINGSPACE will provide: they will be of a wide spectrum, from the more 'object-oriented' to the most 'abstract' ones.

1. The system will allow children to express their ideas if they want with 'entities' that are centred on objects, corresponding to their phenomenological status. The manipulation (change) of each factor/property of an 'entity' will have a visual consequence. These properties which concern real objects could be considered as a kind of *"proto-variables"*, able to evolve to more abstract ones. The *'object-centred entities'*, which represent specific objects, will have various properties, both those that could play a role in the object's behaviour, and others that do not play any role (for instance the colour of a moving object, in a problem studying the motion of the object).
2. A more abstract entity could be considered as a 'construct' depicting an object from a group of uniform real or imaginary objects, that take meaning in the context of a phenomenon, system, process or speculation. This more *abstract entity* represents in general an abstract conceptualised construct that describes the common characteristics of a set of uniform objects. The properties of these entities have a general value that could characterise all the similar objects. For instance, a small circle or a point could represent and model any object that is moving. This more abstract entity is characterised by abstract variables, that are closer to many scientific ones.
3. Abstract entities that directly correspond to abstract *scientific variables* in symbolic form, and that do not have any unique link to a specific object of the

real world, which they may occasionally represent and describe. For instance, staying in the area of mechanics, such an abstract entity, could be the concept and variable of acceleration (expressed with its literal name or symbol *a*).

B) *Use of appropriate symbols, only when these can be considered as socially constructed or accepted symbols.*

Except the considerations on the entities, we have to examine the status of the symbols.

Scientific symbols correspond to socially accepted meanings. Given that young students have not yet constructed the scientific concepts, symbols presented in the books, or in the educational software, cannot be expected to represent for them the social accepted meanings as these were defined by the scientists decades or centuries ago.

One approach could be that the variables should be presented with their literal name, avoiding the scientific symbol, when possible.

Another approach even more appropriate is to allow students to define and use the names and symbols that they want, both for entities (concrete or abstract ones) and properties or variables.

Consequently, an environment that really allows students to express their ideas must offer a great variety of constructs appropriate for this purpose, and ***be open*** to teachers and students to insert ***new entities*** and name these or their properties by a literal or a symbolic mode.

1.4 The representations of executed models and their role

The executable models when running could produce different kinds of representations: phenomenological representation of the whole phenomenon represented by the model as well as various representations of the data related to the variables included in that model.

i) Simulations that correspond to the abstraction level of the corresponding properties or variables: from concrete to abstract ones

Our previous considerations on the status of entities and variables and their differentiation have consequences on the execution of the corresponding models and the produced simulations.

Every modeling system that supports executable models, provides simulations of their evolution.

Each kind of entity, when at least one of its properties is linked with another property of the same or other entity, must be simulated in a way that corresponds to its status and its level of abstraction.

The simulation of object-centered entities that contain proto-variables, must be appropriate one, despite the needs for multimedia material to be used and combined, in order to produce real word simulations.

ii) Representations of data: when and how?

In simulation-based learning environments, in order to explore the behaviour of the phenomena, different representations are provided. Examining all the existent systems of exploratory software, which are based on simulations, we perceived that when the user gives a set of values in the corresponding variables, he/she can run the experiment in order to see the simulation of the phenomenon. When the user wants can ask, either after the simulation has run or at the same time, to see one or more of the available representations related to the last 'run' of the simulation.

Lets examine this situation, using the concepts of "computational transposition" (Balacheff, 1993) and mental operations. When a scientist or the students in the science laboratory have to do an experiment, they have to prepare the necessary instruments in order to take measures, and then to use them to create the corresponding graphs. In the simulation systems, scientists know how the numerical data are produced during simulation and how graphs are created. But, children without a sufficient experience of real experiments, they can create a false mental representation of what happens.

So, answering "when and under what conditions" to provide the data representations: we propose to provide measures and data representations only on demand, after conscious actions and not in an automated way. We could create systems, that when students need some data and their representations they have to ask/do explicitly something, in order to select and represent them. The student must be prevented from asking to select specific data just before the model runs in order to study the corresponding representations. Otherwise, the system may not show them.

The last decade, many researchers argue that multiple representations may have important benefits to the learning process. Researches have concluded that the use of many representations may serve at least two main goals (van Someren *et al*, 1998). They may be used because the information to be learned has multiple characteristics, so that each one may be most adequately conveyed by a specific representation (they support multiple ideas and processes). A second use of multiple representations is to help learners to develop a better understanding of a domain by constraining interpretation that can be made about other representations and the domain to be learnt (Ainsworth *et al.*, 1996). Representations constrain interpretations and promote a deeper understanding of the domain.

The modelling system must provide a variety of data representations, such as: graphs, tables, bar charts, pie charts, etc. in a way that their correspondences may appear, but not automatically (Ainsworth *et al.*, 1996).

1.5 The difficulty of inquiry learning and the importance of self regulation, scaffolding, and metacognitive support: the need for appropriate tools to support metacognition and provide scaffolding

Inquiry learning is labor intensive in any age and class, but especially so in young students and large classrooms.

Even though students in inquiry-based learning are expected to assume a great deal of responsibility in structuring and conducting their investigations they often do not know neither where to begin nor how to proceed. Under such problematic situations, students' self regulation and learning are reciprocally affected: if students do not understand how to do inquiry and if they do not take the time to be more reflective and think about what they are doing and why, then their self-regulation and learning will suffer.

In all the cases scientists agree that the modelling process is a reflective inquiry process.

According to Pintrich (1999), self-regulating learning is "an active constructive process whereby learners set goals for their learning and then attempt to monitor, regulate, and control their cognition, motivation and behavior, guided and constrained by their goals and the contextual features of their whole environment" (p.453).

Self-regulation during problem solving is considered as one aspect of metacognition as defined by Schoelnfeld (1987) and Brown (1987). It includes knowledge about one's own cognitive process, action regulation and control, as well as intuitions and conceptions about the context the activity takes place in.

In order to support inquiry learning and self-regulation two approaches have been applied in technology learning environments:

a) students receive helpful guidance when they need it,

b) students need support to keep track of what they have been doing, so as to reflect on it later.

Even if they are interconnections between them, it is fruitful to distinguish two corresponding theoretical and practical assumptions and aids:

a) This first approach focuses on scaffolding, providing guidelines, prompts and hints

b) The second approach focuses on metacognition, providing tools that help the students to reflect on and analyse their own activity.

a) Scaffolding and associated tools and features

It is argued that for inquiry learning to be successful in overcoming the obstacles that students face during the modelling process (such as organizing and managing data in ill-structured, open-ended science investigations) while assuming the primary role in their own learning, students need to be engaged in reflective learning. One of the most important support in helping students become self-regulated learners has to do with scaffolding, with guidance in deciding the investigation strategy they should use and where to go next. The scaffolded activities are aimed at helping them learn about the characteristics of scientific laws and models, the process of modelling and data analysis the nature of scientific argumentation and proof (Kyza *et al.*, 2002).

However scaffolding is an ill-defined term: the literature abounds with multiple definitions of what scaffolding means, with the boundaries of a definition accepted by the community at large still away from solidifying. This bears with it several implications, one of which being that people who engage in the design of learning environments (software-based and/or curricular) have no unifying core of experiences from which to draw in building their own educational artifacts. The fact that as a community of educational researchers and practioners we have not reached a consensus on how to best scaffold learning has consequences on the extent to which one is able to support student learning and also poses significant problems on how to support designers of such activities.

White and Frederisken (1999) have developed (in Director), the Inquiry Island, that provides a community of software advisors that embody a theory of scientific inquiry and modeling. These advisors guide users as they undertake research projects (independently of the modelling or simulation learning environment which they investigate), with the aim of enabling middle-school students to develop widely-applicable expertise related to scientific inquiry and reflective learning (White, Shimoda and Frederiksen, 1999). Inquiry Island includes task advisors, like the Questioner, Hypothesizer, Investigator, Analyzer, and Modeler, who each provides expertise and advice in their arena of inquiry. It also includes general-purpose cognitive, social, advisors, like the Inventor, Collaborator, Monitor, and Reflector. To facilitate the use and transfer of expertise, advice is presented in domain-independent form (such as, “check to see if your data fit any of your hypotheses”) and includes examples of how the advice could be employed in a variety of domains (including the social, physical, and biological sciences as well as everyday life).

Model-it 3.0, called actually Theory Builder, is designed according to the Guided Learner Adaptable Scaffolding, that supports three types of scaffolding: supportive, reflective and intrinsic scaffolding.

Supportive scaffolding is provided through messages which appear to students when appropriate, guiding them through subtasks of the process, (plan before building, test periodically) or providing examples if needed. Reflective scaffolding is provided by a Notepad that appears alongside the main window, and by description fields. The learner reflects by typing plans, descriptions, predictions, and evaluations into appropriate fields of the Notepad, as seen relevant to each sub-task. Many of the guiding scaffolds refer to text fields prompting the learner to fill them in, when appropriate. It constitutes a passive scaffold. Finally, the designers of Theory Builder, consider as intrinsic scaffolding the different representations of the data provided to the students.

Similarly, according to the environment SimQuest (Joolinger and de Jong, in press) scaffolding during inquiry learning could be supported by the indication of inquiry or modelling typical phases in an appropriate menu, asking students to find correspondences between their applied process phases and the typical ones.

Finally, other researches consider that it is through *Reflective self-assessment* of the inquiry activities that their functional significance becomes apparent to the students (Schon 1983). Often in conjunction with scaffolded inquiry, students are introduced to a reflective process in which they evaluate their own and each other's research. This process employs a careful chosen set of criteria that characterize expert scientific inquiry (being systematic, reasoning careful, etc.) to enable students to see the intellectual purpose and properties of inquiry steps and their sequencing. By reflecting on the attributes of each activity and its function in constructing scientific models, students will gradually understand the nature of inquiry and the habits of thought that are involved.

b) Metacognition and associated tools and features

Metacognition has to do with awareness and explication, with judgment of one's own mental activity, as well as with decision making regarding continuation and self regulation (Noel, 1997).

In order to support metacognition, students need help to keep track of what they have been doing, so as to reflect on it later.

Apart from the difficulty of the modeling process, as this appears in each inquiry activity (discussed in previous paragraphs), it is important to pay attention to the students' difficulties to interpret the models behaviour during its execution.

Previous research on young students' interaction with computerized learning systems have shown that when students observe a simulation, the experiences are not automatically significant at the level of learning effects.

The young and no so young students have difficulties in interpreting the situation, when for instance the result of the simulation is opposite to their expectations (Brown and Burton, 1987; de Jong and Njoo, 1992). They distort the new information in ways that are consistent with the existing cognitive structures. One of the fundamental reasons these distortions are taking place is because they lack meta-conceptual awareness of their beliefs.

If children have awareness of the explanatory and hypothetical nature of their beliefs, they will be more susceptible to ask questions to themselves or to try alternatives ideas that they have or that are presented through teaching (Vosniadou 1994).

In order to promote the development of meta-conceptual awareness and help them in the exploration process, we could invite them in a systematic way to express clearly their initial predictions, their observations during 'running' and their explanations for each experiment. One approach is to invite children to express them orally or in writing in a paper worksheet (approach adopted in DM3 research, Scanlon 1998) or in an electronic one. So, to invite them to write down their predictions, observations and explanations in a special electronic Notebook incorporated in the environment. The advantage of the electronic notebook is that it gives the sense of a unified and indispensable process between the operation of modelling and investigation with the operation to take notes on these actions assumptions, interpretations and changes.

Furthermore, if we take into account that the modelling process is a social activity, the results of the whole modelling activity must be presented and discussed at least in the social context of the class. Students clearly express their ideas to each other or to an audience through writing, drawing diagrams, and speaking so that others understand their research. For this purpose students need to prepare a *report* on the whole process. The report preparation could give them the opportunity to come back and examine their process and explain why some intermediary models were not appropriate.

However, given the difficulty of such a metacognitive analysis, we consider that in order to really support the process of returning to memory, and analysing one's own activity: tools of presentation or even better analysis of history of their process must be available to students.

c) The need for tools to support scaffolding and metacognition

A combination of reasons and requirements for scaffolding and metacognition, lead us to the acknowledgment of the need to design a number of complementary tools:

(i) Promoting a metacognitive mental activity on a part of, or on the whole modelling process involves also to come back and re-examine this process. However, it is very hard to ask from every one to reflect on their process without supporting him/her. In order to proceed, in some significant cases of modelling and problem solving, and try to come back and reflect on one's own process of model creation, a support is needed for one's memory. There is the need for appropriate tools that offer traces of the group's previous activity and support for the groupmemory (the memory of the face to face collaborative group of students).

(ii) In order to promote metaconceptual awareness, there is need to provide students with multiple and flexible tools, so as to facilitate them to write down their thoughts during the different instances of a modelling process:

(ii.a.) **sticky notes**: Sticky notes are the freest form to note something (a short paragraph) down in the working area, in order to specify or remind oneself of something.

(ii.b) **structured notebook**: that could invite students to note their thoughts during each initial analysis, expectation, or observation.

(ii.c) **report** (final report): it is most appropriate to paste there, images from models and data, that will help students to prepare a whole report to accompany the final product/ final model.

All these text annotation tools must be connected with the model and the theme of study, and a synergy between all of them will be supported (copy-paste functionality).

(iii) In order to support reflective inquiry by scaffolding, it is important to create and customize templates that address the specific goals during problem solving. The final report as well as the notebook could provide students with appropriate prompts (depending on their age and the task category) that may scaffold their activity. The scaffolds will be explicitly conceived to guide students and to help them acquire general inquiry skills. Prompts could help students to concentrate on the important points in the investigation. These prompts could be defined with the help of teachers that will work in schools and after some initial experimentations.

2. About Collaborative Learning Support and the Learning Community

The specificities of our project is that it supports different kinds of modelling activities, for different ages of students, and various kinds of problems (from simple to more complex ones, that need a short or a longer period of time to be studied). Synchronous and asynchronous collaboration will be supported in order to take advantage of both modes, and assure flexibility of use. Students as well as teachers are considered as main users in this learning community. The different settings of collaboration as well as the conditions of use will be presented in Part IV of the current document.

The main questions that we try to answer in the next paragraphs are the following:

- Why are we interested in collaborative learning for modelling activities and real school settings? What are their potential?
- How could we coordinate action in synchronous collaboration and how could we support action awareness?
- Why is the written dialogue important and what kinds of dialogue tools we could offer in order to promote flexible but also significant dialogue?
- What kind of collaborative activity analysis tools do we need in order to support teachers and students?
- What is needed in order to facilitate more global exchanges and interactions in the frame of the emerging technology based Learning Community?

2.1 What are the expectations from collaborative problem solving activities?

In ModellingSpace we try to profit from the positive learning potential of all kinds of collaborative settings, related to time and space dimensions: face to face collaboration, synchronous and asynchronous collaboration, through local and broader networks. There are social-order reasons but most important cognitive order reasons for this:

a) Social-order reasons (a work mode of actual life)

To learn to communicate and collaborate is an important skill of the actual life. One of the basic requirements for education of the future is to prepare learners for participation in a networked information society in which knowledge will be the most critical resource for social and economic development. Collaboration is integral to today’s organisations, which require individuals who can work together to solve complex problems and share their own knowledge and expertise with others. Collaborative skills can be learned, and it is therefore essential to provide individuals with appropriate learning opportunities (Constantino-Consalez and Suthers, 2001). It is considered that if students remain in social and intellectual isolation, students may fail to develop and refine those cognitive and interpersonal skills increasingly necessary for social and professional life (Abrami, 1996).

b) Cognitive reasons

Studies of the achievement effects of cooperative learning have taken place in every major subject, at all grade levels. As a result of this research there is a growing consensus among researchers about the positive effects of cooperative learning on students’ achievement (Slavin, 1995; 1997) under appropriate conditions, i.e. with appropriate tasks and in specific collaborative settings. Unlike the teacher-centred models the principles of cooperative learning are based upon a learner-centred model that treats the learner as an active participant. The conversations, multiple perspectives and arguments that arise in cooperative groups may explain why collaborative groups facilitate greater cognitive development than the same individuals achieve when working alone (Harasime, 1997).

We could distinguish three main reasons that underlie the achievement effects of cooperative learning: motivational, social cohesion related reasons and cognitive elaboration reasons.

- ***Motivational reasons***: These take into account the reward or goal structures under which students operate. Cooperative incentive structures create a situation in which the only way group members can attain their personal goals is if all the members of the group are successful. In these conditions, group members must both help their groupmates to do whatever helps the group to succeed and to encourage their groupmates to exert maximum efforts.
- ***Social cohesion reasons***, based on the idea that students help their groupmates learn because they care about the group. The social cohesion perspective emphasizes team building activities in preparation for cooperative learning, as well as group self evaluation, instead of external incentives and individual accountability. A well known application of this interpretation is Aronson's Jigsaw method (Aronson, Blaney, Srephan, Sikes and Snapp, 1978), where students concentrate on different topics in 'expert groups' and subsequently share their expertise in groups. The theoretical idea in a Jisgaw method is to create interdependence between members in a way that would increase social cohesion. Therefore, distribution of a task among several agents has fundamental cognitive significance.
- ***Cognitive reasons:*** collaborative learning is assumed to be effective because it requires participants to elaborate their cognitive structures in a social context.

  Miyake (1986) and Hutchins (1995) have argued that social interaction (and interaction with the tools of technological culture) provides new cognitive resources for human cognitive accomplishment. In a shared problem solving process, agents who have partial but different information about the problem in question, all appear to improve their understanding through social interaction. Through social interaction, the <u>contradictions, inconsistencies and limitations of an agent's explanations become accessible, because it forces the agent to perceive his or her conceptualisations</u> from different points of view. Limited cognitive resources can be overcome by distributing the cognitive load to several agents, each of whom is equipped with a restricted power of cognition. Externalisation is an important prerequisite of socially distributed cognitive achievements.

  A fundamental assumption is that interaction among children around appropriate tasks <u>increases their mastery of critical concepts</u>. Both major traditions on developmental psychology, the Vygotskian and the Piagetian have substantially contributed to the theory of collaborative learning. Particularly, Vygotsky's idea of the zone of proximal development has been useful for understanding mechanisms of collaborative learning. According to this view, collaborative activity among children promotes growth if children of similar ages have developmental differences.
- Similarly, research on self explanation effects, has revealed that explaining problems to own self fosters cognitive achievements. Hatano and Inakagi (1992), as well as Brown and Palinscar (1989) have argued further that deep conceptual understanding is also fostered through <u>explaining a problem to other inquirers</u>. In order to explain one's view to one's peer, a student has to cognitively commit himself or herself to some ideas, explicate his or her beliefs, as well as organise and reorganize his or her knowledge. Through this kind of process inadequacies

of one's understanding tend to become more salient. It is considered that one of the more effective means of elaboration is explaining the material to someone else. Several studies of peer tutoring have found achievements benefits for the tutor as well as the tutee (Devin-Sheehan, Feldman and Allen, 1976). Webb (1989, 1992) found that the students who gained the most from cooperative activities were those who provided elaborated explanations to others.

Moreover in cases of problem solving in rich and critical conceptual domains it appears that collaboration through a network could be more effective than face to face collaboration, specially due to potential cognitive effects presented above.

It seems that for purposes of a communication aiming to conceptual change, written communication, combined with face-to-face communication, is more effective than face to face alone because it requires more extensive thinking process (Cohen, 1994).

During collaboration through networks, the need to externalize one's own thoughts, in a written way, has significant effects. Specially, when the learning activity implicates rich conceptual knowledge that is under development.

In general, in cases of knowledge seeking inquiry, technologically sophisticated collaborative learning environments designed to follow cognitive principles could provide advanced support of distributed process of inquiry, facilitating advancement of a learning community's knowledge as well as transformation of the participants epistemic states (Pea, 1993). Moreover, social interaction fosters the emergence of a more abstract conception, than individual working (Schwartz, 1995).

Summarizing, the above considerations, we consider that collaborative learning during modelling activities could significantly support knowledge development and construction, at a conceptual level as well as of the modeling and inquiry process. The considered potential cognitive effects, give as indications of the value of different collaborative settings such as: *Peer tutoring, collaborative learning based on distributed tasks, self explanation and learning by examples, etc.*

2.2 The importance of appropriate means for actions' coordination and workspace awareness

Coordination of actions: During collaborative learning a common final product is expected from the participants, so a shared workspace must exist. In this way the learners have a shared point of reference. All the participants must have access to the common workspace; as a result, the production of the final product must be coordinated.

Among the various ways for *regulating the action access during synchronous activity*, the *Action Key approach* (the student that has the Action Key controls the common workspace) appeared to be an appropriate choice, applied in the system COLER, developed by Soller (Soller, 2001), given that it provides clear semantics of actions and temporal roles during activity. Action key approach was tested in HCI lab (University of Patras) with similar activities of concept mapping and using the Representation tool, which has common features with the under design ModellingSpace environment (see, Feidas, Komis & Avouris, 2001, incorporated in Appendix B). The metaphor used is that of "passing the key". The holder of the "action-enabling key" is the temporally active partner on the level of actions. Through this key request/ key accept/ key reject protocol the active role can change at any point during collaboration, provided that the passive partner requests the key and the active partner accepts the request. An implication of this "key exchange" protocol is that deadlocks can be created in cases when one partner cannot proceed with problem solving and at the same time refuses to pass the key over to the other partner. Despite this, the protocol maintains clear semantics of actions and roles in the shared activity space. This conclusion seems to be in agreement with the view expressed by researchers of similar environments (see Soller, 2001).

During a supervising synchronous activity, teachers must consider the case of a possible deadlock, if the student does not give the Action Key to the other learner that ask for it.

Another question concerning the coordination of actions is related to the *rights that each partner has on the contributions of the other* partner. In order to answer this question, an experiment was organised, in HCI laboratory (Fidas, Komis, Avouris, Dimitracopoulou, 2002, see Appendix A). Two alternative collaboration protocols were used; Groups (A) had no ownership control while groups (B) maintained ownership of introduced objects, so partners were not allowed to modify objects introduced by their peers.

In the case of groups (B) every time a partner needed to modify an object of different ownership, a negotiation phase had to be initiated concerning the purpose of the modification, in order to convince the object owner on the proposed modification. This ownership control mechanism was effective in inhibiting modifications of created objects by other partners. In contrary, the group A without ownership control, disagreeable instances during collaboration happened when a partner started deleting objects in the common space, and the owner sent angry text messages

It was observed that the dialogues of group B were longer, since one partner requesting a modification of somebody else's part of the solution needed to negotiate the modification first, resulting to significant rich dialogues. The dialogue may be semantically rich.

However, discussing further the above mentioned results, we could consider that eventually students need a clear indication of the ownership (with the direct or indirect indication of names of owners, in each item of the solution), in order to regulate their activity and avoid this kind of conflict.

Workspace Awareness: As presented in section 5.4.3 (Part II), *workspace awareness, during synchronous collaboration* concerns the up to the minute knowledge a student requires about other students interactions with the shared workspace (What are the other members of the group doing to complete the task? Where are they? What are they doing? What have they already done? This awareness is essential if the students are to learn and work together effectively. Collaborating learners maintain this awareness by tracking information such as other learners' locations in the shared workspace, their actions, the interaction history. Workspace awareness is necessary for effective collaborative work, but also plays an integral part in how well an environment creates opportunities for collaborative learning.

So the system must display information such as:

- who is participating;
- who is controlling the share workspace;
- what is the contribution to the current product for each student (for instance it could be shown by the names or colour coding who has introduced what, either if it is 'items in the modelling area, or utterances in the various text based tools);
- what each student is doing (e.g. waiting for the Action key).

Concerning the control of the shared workspace, a control panel may be used, to show the activity of the pointing device, e.g. it is highlighted when dragging and moving of objects occurs by the collaborating partner. This mechanism can reduce the ambiguity observed in turn taking (Feidas, Komis, Avouris, 2001, see Appendix E)

During asynchronous collaboration, workspace awareness, refers mainly to the product history, thus to the question: what is the contribution to the current product for each partner? Concerning the design of models, the names of contributors or coded coloured cards, could appeared under demand. Concerning text based contributions different colour coding could be sufficient to fulfil

2.3 The importance of appropriate means of dialogue

Interactive linguistic exchanges between people play an essential role in the elaboration and perpetuation of scientific concepts. This is for several reasons and on different levels:

- Scientific concepts are not to be found in nature, they are products of scientific culture; and this cultural heritage despite being largely expressed in written language, is kept alive in situations that involve linguistic interaction (Vygotsky).
- Conceptual acquisition and change may occur in children as a result of interaction with concrete or abstract 'objects'. Objects' manipulation in didactic situations can be designed to give appropriate feedback: the meaning of the feedback depends on explanations and negotiations mediated by language.
- Concepts are intimately associated with signs, of which they constitute the signifiers within sign-systems such as languages, of which the primary use and mechanism of acquisition is social interaction (Bakhtin).

Externalisation through a written way, during collaborative activities may have significant effects, specially for the kind of learning activities, that are conceptually rich, as is the case of the activities that will be applied in the frame of ModellingSpace project.

Most systems offer one or two means of dialogue, for instance, chat and/or structured dialogue interface. Given the discussion on the advantages and disadvantages of various dialogue tools (see section 5.4.2 in Part II), we consider that it is important to provide to students multiple tools of dialogue, so as to assure flexibility of use, in different instances, according to the apparent needs in different phases of problem solving, and according to the needs that derive from the complexity of the task.

Lets consider the case of synchronous collaboration. It is needed to provide three different dialogue modes:

a) **Chat tool** (unstructured, synchronous dialogue): it is appropriate during initial brainstorming phase of problem solving, discussion on problem solving or modelling strategy, eventual decisions of the distribution of the task to different members.

b) **Structured chat interface**, with specific sentence openers: Providing the tight degree of constraint on typewritten communication can in fact promote more focus on reflection and the fundamental concepts at stake. Problem solving and interaction tasks need to be more interwoven via computer supported communication in order to allow the emergence of a variety of forms of cooperation. Students have the tendency to be drawn into the graphical task, to the detriment of communication. The chat box interface enforces this separation while structured interface improves students' collaborative argumentation because they are encouraged to use certain discourse acts and problem-solving activities. As argued in section 5.4.2 (Part II), the most suitable approach to structure the dialogue, and specially for problem solving collaborative activities such as these supported by ModellingSpace, is sentence openers. It is critical that these openers enable the widest possible range of communications with respect to the learning task The openers must originally develop from an analysis of face-to-face collaborative learning of the same task and environment, and should be refined several times to accommodate users' feedback and observed experiences. In this process we will exploit the work and the results of others who have worked in similar tasks (for instance, this of Soller, 1999). Finally, it is to be noted that, structured dialogue holds crucial benefits for significant meta-analysis of collaborating students.

c) **Sticky notes**: They constitute a freer form of expression and provide a way for the users to specify with more accuracy the 'objects of discussion' (Kyza *et al.*, 2001,). They have the advantage that it is easier to refer to parts of the artefact, and thus help to recover the portion of the discussion that is concerned with a given part.

Sticky notes are referred as *'embedded discourse'*, given that they embed comments directly on or in the display of the artefact under discussion. Some empirical evidence supports this embedded approach of discourse (Guzdial, 1997; Wojahn, 1998). In cases of separate artefacts there is a greater distance between the object of the discussion and the annotations, and hence cognitive load in processing them. In the experimentation of HCI lab (Appendix E), the existence of sticky notes allow students to have more specific and clear discussions.

It is to be noted that the disadvantage of typical sticky notes include the fact that the record of discourse is fragmented across the artefact, making it more difficult to get a sense of the whole discussion or to notice relevant relations between discussions about different parts of the artefact, as well as the possibility that the artefact becomes cluttered with comments.

We would like to be able to recover chronological versions of the discourse and perhaps to index the discourse in different ways other than by artefact component or by chronology (Suthers, 2002).

Researchers propose to conceive *linked dialogue representations* between sticky notes and chat or structured chat, so that someone can see the one or more sticky notes in the chronological history of dialogue.

Lets consider the case for Asynchronous collaboration:

During modeling activities and problem solving tasks, we do not consider e-mail as the appropriate means of communication. The written communication could be supported mainly, through the available text tools:

- sticky notes
- structured notebook,
- and reports

under the condition that they keep track of each partner's contribution (with color coding), as well as the chronological order.

Other designers and researchers offer e-mail as a means of communication. In our project context, threaded discussions through e–mail could be useful only during general discussion on general topics concerning modelling activities that take place among an important number of discussants (more than two), via the server.

2.4 What is the role of the teacher during collaborative learning? What kind of collaborative activity analysis tools do we need, in order to support teachers as well as students?

Students naturally seek the teachers' help when they realize that more information is needed in order to benefit for the continuation of an interaction. Teachers are also socially responsible for students' learning. A crucial actual question is how we could help them to fulfill those responsibilities in computer based collaborative situations?

Three main questions are raised:

- How can teachers help students?
- How can teachers be supported by appropriate tools to help students?
- Moreover, how can teachers learn as a result of helping students?

Little research has yet been carried out on the possibility that teachers could have a significant role during collaborative learning through network, and that they can derive useful knowledge from observing or participating with their students in CSCL environments (Lund and Baker, 1999). Some research has focused on the kind of teachers' interventions, and not on how we could support teachers to carry out these interventions or on their needs during tutoring or coaching of collaborating students.

We consider that most of the existing collaboration systems present limitations when used by young students in real school settings. Some of the limitations are attributed to the fact that the teacher who is in charge of several students, fails to interpret the enormous number of complex interactions that can take place simultaneously.

In order to examine the needs of teachers during synchronous collaboration, and determine requirements that could be relevant for ModellingSpace project we have carried out an investigation.

The question was to examine how teachers' and students' behaviour during synchronous problem solving with known and current learning activities (and not innovative ones) (see short report Petrou, Appendix D). Ten students were separated in two groups working in two different PC laboratories in the same building. Students had to resolve simple and short problems (related to programming in Pascal) in usual school conditions in 45 minutes of a one hour class.

We tried out two possible scenarios for teachers' interventions:

a) On-line supervision of group of students collaborating in a synchronous mode.
b) Off-line analysis of preceding intervention: the teacher studies the students dialogue and action-based interaction, then during the next session (next day or two days later) intervenes in order to discuss some concepts, or in order to propose new problems.

In the teachers' workstation Netsupport software was installed to allow them to supervise students' screens via the local network. The thoughts of teachers about the advantages of this kind of work are reported in the Appendix. Their needs concerning the way to support collaborating students were among others the following:

- Teachers mentioned that class supervision was necessary. About how many students' groups they would be able to supervise simultaneously, they estimated that they would not be able to supervise more than 4 or 5 groups of students in parallel. Eventually more sophisticated modes of supervision could allow them to be more effective in this task.
- During supervision, they need to have a clear indication of receipt of a message (question from a group of students), given that they may be distracted from other activities or other groups of students.
- They expressed the need to examine the history of the interactions. Teachers that examined the logfiles of interaction were impressed from the conceptual difficulties that these could reveal, difficulties not apparent during the oral discourse with the whole class.
- For the intermediary or final products of a problem's solution, as these appeared in the shared working area, they wished to see what was the contribution of each partner. Similarly when someone deleted an item, to be able to see who deleted it.
- They thought that they could not estimate the contribution of each student during problem solving given that amongst peers of students one often takes the active role in the workspace and the other the evaluator's role, writing messages, evaluating the process, actions and solution, and thus guiding the other. Both roles are important. In the case of the latter student, logfiles of actions or of the state of the working space do not reveal his/her contribution.
- They expressed the need to be able to reconstruct the state of the shared workspace for each time lapse, while studying the dialogue history, in order to be able to interpret what had happened.

- They also mentioned that, if there was a way to analyse and present in an appropriate mode the whole interaction, this would have helped them, to intervene or give advices and hints at the end of the session or during the session.

Summarising, the most important requirements, teacher need:

a) a way to supervise multiple groups of students that collaborate in a synchronous mode;

b) an appropriate and easier mode to take profit from the detailed logfiles of students' interactions;

c) if possible, an elaborated mode of analysis in order to examine in few minutes the whole history of interaction.

Consequently, we need to design and develop:

a) supervising tools and facilities;

b) tools that are based on the logfiles, make them eventually more easily readable, and provide a link between dialogues' history and state of the common workspace;

c) tools that produce an automated analysis of students' interaction.

It appears that the most difficult requirement to accomplish is the third one.

It is just recently acknowledged that a new research direction should be concerned with how we could take profit from the traces/transcriptions of students so as to facilitate the teachers' analysis task (Baker, 2001) and support diagnosis and scaffolding.

In order to develop effective analysis frameworks and tools for collaborative problem solving analysis, we need to investigate some key questions:

- How to coordinate the analysis of actions and dialogues?
- What are the most significant data to be logged and coded?
- How to inter-relate collaboration features with problem solving content and process and what abstraction methods we need in order to construct a computational model?
- How could these transcripts be structured?
- How to provide a rich variety of analysis output, to assist facilitators or experienced learners?

In order to answer these questions, that appear crucial to our consortium and given the complete lack of related research, we have worked on this topic during the first period of the project lifecycle. The reader is invited to read related papers Avouris *et al.* (2002) (see Appendix B), and Dimitracopoulou *et al.*, (2002) (see Appendix C).

A framework of analysis was developed that could support the development of appropriate analysis tools for students' interactions. The 'Object-oriented Collaboration Analysis Framework' (OCAF), identifies patterns of interaction and relates them to objects of the shared solution. The corresponding model provides a new way of representing collaborative problem solving activity, taking into account both actions and dialogues of partners and supporting qualitative and quantitative representations that can be used as meta-analysis and evaluation tools.

The proposed framework is based on two basic considerations, one related to the *object oriented view* of collaborating actors' roles and contributions and the other to the *unified analysis of dialogues* and *actions on objects*.

The entity/relationship/attribute constructs could be one set of the basic objects that make a diagrammatic solution. The proposed model according to OCAF has been formalized in textual and diagrammatic form as follows:

Let a given Solution S of a problem X be: S(X) = { E_i , R_j, A_m,} , Where E represent the node entities of the solution, (i=1, …, k) R the relationships connecting them (j=1,…,l) and A the attributes of the entities (m=1, …, n) that participate in the solution.

The model of the solution can be:

$$M\ (S) = \{\ E_i\ \ {*}\ \tau i\ /P_i\, f_j,\ P_k\, f_l\ ,\ \dots\ \ R_j\ \ {*}\ \tau i\ /\ P_i\, f_j,\ P_k\, f_l\ ,\ \dots\ ,\ \ A_m\ \ {*}\ \tau i\ /\ P_i\, f_j,\ P_k\, f_l\ ,\ \dots\ ;$$
$$-E_i\ \ {*}\ \tau i\ /\ P_i\, f_j,\ P_k\, f_l\ ,\ \dots\ ,\ -R_j\ \ {*}\ \tau i\ /\ P_i\, f_j,\ P_k\, f_l\ ,\ \dots\ ,\ -A_m\ \ {*}\ \tau i\ /\ P_i\, f_j,\ P_k\, f_l\ ,\ \dots\ \}$$

Where: E, R, A, are the entities, relations and attributes that are part of the final solution, while with –E, -R, -A the items discussed during the problem solving process, but not appearing in the final solution, are shown. τ_i is an index of the item, as implied by its initial action of insertion or by its discussion in the timeline of the problem solving process.

To each item a sequence of $P_i f_j$ is associated. Each $P_i f_j$ represents the human agent Pi (e.g. a student, a teacher or facilitator) participating in a direct or indirect way in the problem solving process and his/her functional role f_j related to the particular part of the solution.

The different functional roles f used in OCAF are described in Table 2.1. It should be noticed that two functional roles concern the initial proposition to insert the item (by action (I) or by dialogue (P)), while the others express the discussion on each item. Also testing of the proposed solution is done through argumentation (A) in the case of static-diagrammatic solutions, while testing can involve use of alternative representations and provided testing tools in case of development of dynamic models of the solution (T).

ID	**Functional Role**	**Derived from :**	**Example**
I =	Insertion of the item in the shared space	*action analysis*	*Action:* 'Insertion' of Entity "Velo"
P=	Proposal of an item or proposal of a state of an item	*dialogue analysis*	*Message:* "I believe that one entity is the firm 'ABC'" or "let us put the value of entity flow to state *locked*"
C=	Contestation of the proposal	*dialogue analysis*	*Message: I think that this should be linked to the entity B by the "analogue to" relation*
R=	Rejection / refutation of the proposal	*action and/or dialogue analysis*	*Message:* "What their attributes will be ? I don't agree". Or *Action:* 'Delete' Entity "Velo"
X=	Acknowledgement/ acceptance of the proposal	*Action and / or dialogue analysis*	*Message:* "That's right" *or* *Action: Insertion of* a proposed enitity
M=	Modification of the initial proposal	*action & dialogue analyses*	*Message:* I suggest we put the state to "unlock" *Action:* "Modify"
A=	Argumentation on proposal	*dialogue analysis*	*Message:* "I believe that I am right because this is …"
T=	Test/Verify using tools or other means of an object or a construct (model)	*actions & dialogue analyses*	*Message:* Let us run this model to observe this part of the model behavior *Action:* Activate 'Graph Tool' , or 'Barchart Tool''

Table 2.1: Unified "functional roles" definitions

So for example: [E (CLOCK)]=A_P B_M A_I] indicates that the entity 'Clock' has been produced from interaction of Agents A and B. Agent A made the initial proposal (A_P), which was modified subsequently by Agent B (B_M), finally Agent A inserted the object in the shared Activity space (A_I), accepting the modification.

In our context, a functional role reports the purpose of a 'communicative act', from the point of view of its 'actor' or 'interlocutor', thus constituting an interpretation of the actors/interlocutors intention in communication It is to be noted that, the term of communicative act refers not only to messages (written dialogues during collaboration by distance), but also to actions of collaborative agents.

The diagrammatic representation of the model is easier for the student to study. The figure 2.1 presents two diagrams ('a', 'b'), that model the collaborative history of solutions provided by two different groups of students, referring to the same problem. Notice that the discontinued lines present the items discussed during problem solving but not present in the final solution.

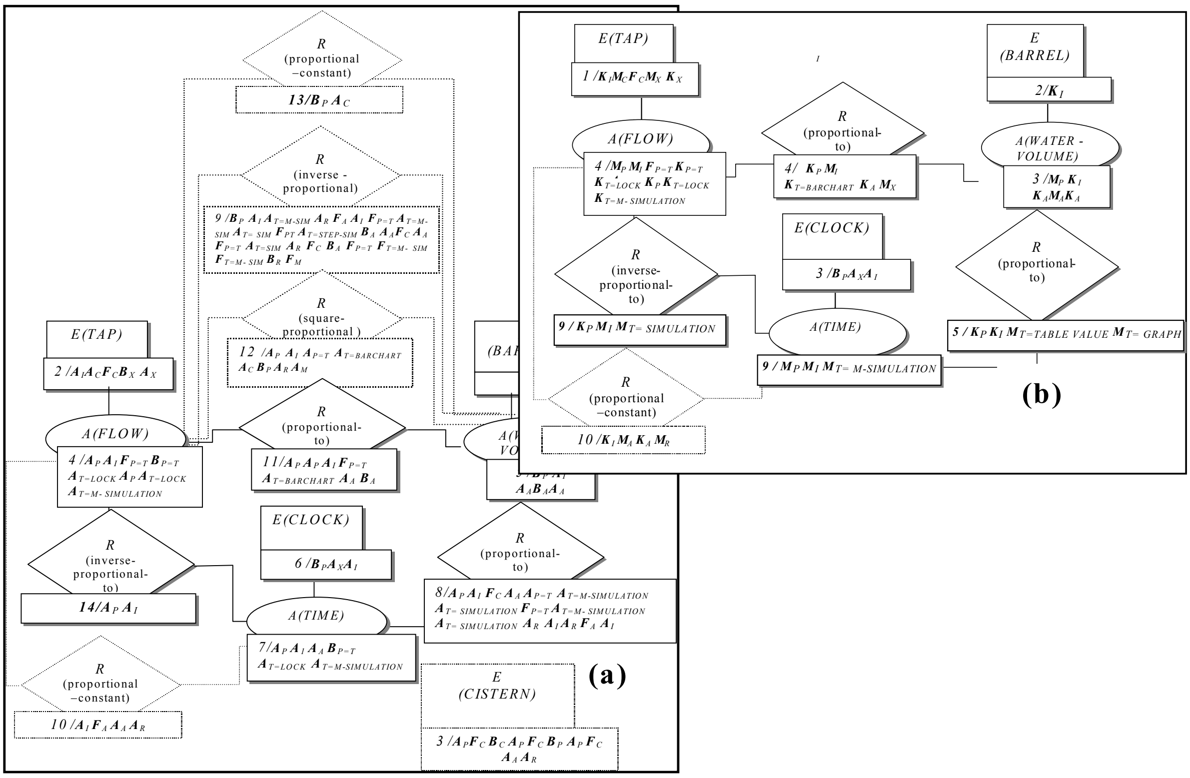


Figure 2.1: Diagrammatic models of two group solutions history (students A,B and students K,M) on the same problem

The value of this kind of model is mainly related to its capacity to bring up interesting points of view and thus provide information mainly to researchers and teachers relating to the quality of the problem solution and the collaboration modes adopted by the participants, such as the following ones:

A) *Analysis oriented to the quality of the solution*: A constructed OCAF model, first of all, provides information on solution items that take part of the solution. Additional information that can help to interpret the solution and eventually to infer students difficulties on conceptual or problem solving aspects are: (a) *Items discussed and rejected* and items that were abandoned due to a conflict. (b) The collaborative history of objects, that for instance can help to distinguish which *non-appropriate solution items* have been *derived from low collaboration*, (c) The order ('τ') of each item discussion could provide *indications on conceptual difficulties* of participants. (d) Information on the problem solving strategy can be extracted by the study of some *'functional roles'* in objects' solution history related to *solution testing* approaches and especially to tools used.

The diagrammatic form of the OCAF model, contributes in a supplementary way to the analysis, providing a perceptual view.

A teacher that examines and compares two diagrammatic OCAF models of solutions, can directly distinguish, for instance, solution objects that are not appropriate and were not discussed in a group, or others that were discussed a lot and revised. Such information can support teachers to propose intra-group collaboration in order to discuss specific issues. Teachers could easily identify conflicting points, or not appropriate approaches and give advice on topics in the debriefing session internal to the group. They could also recognize semantically significant differences between approaches on problem solving and advise further intra-group discussions.

B) *Analysis oriented* to *collaboration modes adopted by participants*: Information that can be derived from queries on a constructed OCAF model can concern among others the following:

- *Degree of participation of group members*, based on indicators such as distribution of solution items per member:
- *Contribution of group members* to the developed solution, determination of *roles of group members* and determination of the degree of their involvement: Regarding the functional roles of each member contribution, it can be accounted, for instance, the distribution of 'items proposals' (functional role 'P') among the agents participating that provides an indication of ownership and involvement.
- Existence of some *significant functional roles for problem solving approach*: It is often important to examine closer the degree of existence of some functional roles, for instance these of 'Argumentation' or 'Test', that help to distinguish some strategies concerning the evaluation process of the produced solution in each specific collaborative environment. In the example presented in Figure 2.1a, it was derived that the pupils had tested parts of the solution (e.g. the relations) by using mostly the available manual simulation (Tool: M-SIMULATION) and had not validated the overall model. Examining the indices of T(est) role, it was derived that only some of the available alternative representations (graphs, bar-charts, tables of values) have been used, and this in a limited degree.
- Identification of *interaction patterns* per item of solution and *Density of interaction*.

Some of the above analysis dimensions are related to quantitative aspects of interaction, and appear often in studies of collaborative distance learning environments, while others relate to a more cognitive and meta-cognitive view, as for instance is the case of solution validation strategies.

Consequently, it is considered that during synchronous collaboration, ModellingSpace must support teachers with the following:

1. Supervision of students' screens, using appropriate tools and facilities
2. Tools that make logfiles eventually more easily readable, and provide a link between dialogues history and state of the common workspace. They are important for teachers that want to diagnose students' progress and their conceptual or strategic difficulties.
3. Tools that produce an automated analysis of students interaction, presented in a diagrammatic form based on the OCAF framework. It is to be noted that this analysis is most detailed and significant when student uses structured dialogue interface.
4. Queries that could exploit furthermore the previous analysis.

During asynchronous collaborative mode, teachers could use:

1. Automated analysis inspired from the OCAF framework. However this must be further investigated.
2. Documents and texts of the students (structured notebook and report) that could retain and show the history of the document

2.5 Researching metacognition support and self-regulation for students in collaborative problem solving

Researchers have focused more on self-regulation by students themselves, instead of on developing tools for teachers. The reasons for this are various, among others that research has not been class focused and some of the research on self-regulation has taken place in laboratories and not with young students. There are some main assumptions, underlying the tools supporting self-regulation of the collaborative activity.

Does the concept of metacognition scale up to a group of individuals who solve a problem, given that metacognition is described as a mechanism in individual problem solving? Nickerson asks this question in the context of the distributed cognition approach (Salomon, 1993). This approach considers group of persons and the tools they use as a single cognitive system and the question is if such a system also has metacognitive skills.

Interaction regulation consists of organizing work inside a group by defining roles or assigning sub-tasks to participants. The hypothesis is that simultaneous regulation of task and interaction is more efficient than regulation of the task alone because it leads to a better coordination of actions. The hypothesis is that in collaborative problem solving, metacognition not only covers strategic reasoning related to the task but also reasoning related to the interaction itself. The hypothesis underlying this work states that regulation of the interaction and regulation of the task are closely related mechanisms and their co-occurrence facilitates coordination.

Actually, some systems have been developed that incorporate tools that reflect interactions, collecting raw data in log files and displaying it to the collaborators. Furthermore, there are systems that monitor the state of interaction, modelling the interaction and providing collaborators with visualizations that can be used to self diagnose and self regulate interaction. These visualizations typically include a set of indicators that represent the state of interaction, possibly alongside a set of desired values for those indicators. The hypothesis is that visualization structures of students' discussion and actions with the aid of a suitable representation can assist students be aware of other's actions and opinions.

It is useful to examine briefly four kinds of systems that present visualizations that mirror the activities of pairs:

Jermann (2001) presents visualization tools, so-called "interaction meters", that represent the number of contributions related to the discussion and to the implementation of the solution. Subjects are presented with a constantly updated visualization of their participation in talk and task related actions through bar charts that show the number of messages and the number of problem solving actions. The design rationale of interaction meters is that they might give subjects a better representation of their participation as well as of the role they play in the problem solving process. According to Jermann, interaction meters reify participation and work organization, and the hypothesis is that these help students build and maintain a more accurate model of interaction. Two types of interaction meters were designed. The first compares subjects by representing their participation side by side as two bars (comparative condition). The second represents participation cumulated across subjects, i.e. one bar chart represents the sum of the subjects' contribution to discussion and another bar represents the sum of the subject's problem solving actions (cumulated condition). These assumptions are tested experimentally with a "traffic simulator" task and software. The results concerning estimation of participation in problem solving actions were compatible with our hypothesis suggesting that the comparative version of the interaction meters is more helpful than the cumulated version and than the absence of interaction meters. The results show that co-occurrence of task and interaction regulation allows quicker solving of the problem, thus better performance concerning the given task.

Moreover, Zumbach, Muhlenbrock, Jansen, Reimann and Hoppe (2002), are in the process of developing an application for co-constructive tasks with functions for tracking, analysing, and feeding back parameters of collaboration to group members, presented by pie-charts.

Simmoff (1999) proposed an interesting way to merge the graphical representation of participation rates and the potential of learning. His system visualizes discussion threads with nested boxes. The thinness of the box edges represents the number of messages produced in response to the opening message for a particular thread. In an educational environment thicker box might mean deeper conversation, hence deeper learning.

Finally, in situations where more than two persons interact, social networks may be used to represent the exchange patterns among participants in a discussion (Nurmela, Lehtinen and Palonen, 1999). A social network typically consists of a network of nodes in which each node represents a participant. The thinness of an edge connecting any two nodes represents the amount of the discussion between two participants.

The two first kinds of visualizations could be closer to the activities of ModellingSpace, given that they concern problem solving during synchronous collaborative activity. But a main difference rests on the fact that the researches have tested these approaches with low-level conceptualization problems. The above methods appear to us to have a low potential to contribute to students awareness, metacognition and self-regulation of their activity.

It is to be noted also that collaboration quality depends on the type of the collaboration, so different indicators must be experimented for different collaborative settings and different ages.

The question must be further investigated after the first experimentations with students, in order to propose tools that are significant for the task and more important for young students in order to really support them and not to add an additional high cognitive demanding task.

2.6 What it is needed in order to facilitate more global exchanges and interactions in the frame of the Technology based Learning Community on modelling that could emerge?

Donald Norman's book, "Things that make us smart" (Norman, 1993), shows how technology does not simply improve the way we do things, but actually changes what we do. In parallel the view of learning as a social practice and as situated learning in communities of professional practice does not serve school-based learners. A key educational implication of situated learning and of the sociocultural approach to understanding teaching and learning actions has been to set the goal of providing students with participation to authentic work of communities of practice.

Lessons learned from a number of research studies suggest that we need to also consider the school as a community of practice. Schooling can be improved by understanding the practices of its participants and by creating environments and systems to help the school be a learning organization. We need to create systems that allow people to act according to their capabilities, and that amplify, transform, and extend their work to new or additional outcomes. Brown and Duguid (2000) argue that information driven technologies and their implementation need to be grounded in the social life of the school rather than in the information space.

Given that most of the schools of the ModellingSpace project will not have a long history in the exploitation of ICTs, our objectives are not very ambitious, given that all the experimentations will take place in a duration of less than eight months. Thus, it is important to provide simple and powerful tools and services.

How can a large group of a community interact and be mobilized to achieve common goals so that the collective effort is greater than what could be accomplished by isolated individuals or isolated groups? The system must allow one to find or post new materials, ask questions discuss, or create workgroups.

One important feature of the learning community on modelling is the **Repositories**:

Repositories, with indexed archives can be provided not only for storing and accessing intermediate or final products, such as themes of study, models, reports, but also to store processes or action histories.

It is desirable to support the sustainability and reusability of the work done, and store it in a server. Information access functions and 'indexes archives' are usually confined to the products of the system itself, i.e. there is no real interoperability to share knowledge elements with external resources (i.e. by exchanging elements with a general corporate memory). So the main problem is that these repositories must be open or easily extensible.

Media repositories: Independently of the products developed with the specific tool/environment, repositories of external materials in common formats (text, graphics, sound, animation) could be provided. For this, good retrieval functions are of particular importance.

It is also important to support the **Awareness** of last actions, providing in a semi-automated way information of all new materials added or modifications and actions performed.

Additionally, it is important to assure a flexible access to these materials: teachers and students could retrieve their material either from school or from home, every time and from everywhere.

Another important feature is **group forming**: Each group has different rights and privileges. For example, in a class group students can not deleted a document created by a teacher or another group, whereas, all members have equal rights and responsibilities regarding the managing and editing of their files and models. They can define their material as public or internal to the group. A flexible approach to create groups and define groups for either synrhonous or asynchronous collaboration must be supported.

Finally, logfiles of all these activities must be recorded in order to examine the exchanges amongst the members of the community.

3. References

Abrami, P.C. and Bures, E.M. (1996) Computer Supported Collaborative Learning and Distance Education, Reviews of lead article. *The American Journal of Distance Education, 10* (2): 37-42.

Ainsworth, S., Wood, D and Biddy, P. (1996) Co-ordinating multiple representations in computer based learning environments, *Proceedings EuroAIED*, pp. 336-341, (Lisbon: Portugal).

Aronson, E., Blaney, N., Srephan, C., Sikes, J., and Snaap, M. (1978) *The Jigsaw classroom.* (Beverly Hill, CA: Sage).

Baker M., de Vries E., Lund K. and Quignard M. (2001) Computer-mediated epistemic interactions for co-constructing scientific notions: Lessons learned form a five-year research program. In P. Dillenbourg, A. Eurelings and Hakkarainnen K. (Eds). Proceedings of *European Perspectives on CSCL*, Maastricht, March 22-24, 2001.

Baker, M., Hansen, T. Joiner, R. and Traum, D. (1999) The role of grounding in collaborative Learning Tasks. In P. Dillenbourg (Ed) *Collaborative learning: Cognitive and computational approaches.* (Amsterdam: Pergamon).

Balacheff, N. (1994) La transposition informatique. Note sur un nouveau problems pour la didactique. In M. Artigue *et al.* (Eds.) «*20 ans de Didactique des Mathematiques en France*», Grenoble, Editions la Pensee Sauvage.

Bliss, J. (1994) From Mental Models to Modelling. In H. Mellar, J. Bliss, R. Boohan, J. Ogborn, C. Tompsett (Eds). *Learning with Artificial Worlds: Computer Based Modelling in the Curriculum.* (London: The Falmer Press).

Bliss, J., Ogborn, J., Boohan, R., Brosnan, T., Brough, D., Mellar, H. (1992) *Tools for Exploratory Learning Program.* End of Award Review Report, London, University of London.

Brown, A. (1987) Metacognition, Executive Control, Self regulation, and other more mysterious mechanisms. In F.E. Weinert and R.H. Kluwe. *Metacognition, motivation and understanding.* (Hillsdale, New Jersey: Laurence Erlbaum Associates).

Brown, J.-S. and Burton, R. (1987) Reactive Learning Environments for Teaching Troubleshooting. *Advances in Man-Machine Systems Research*, Vol.3, pp. 65-98.

Chi, M.T.H., Feltovich, P. and Glaser, R. (1981) Categorisation and representation of physics problems by experts and novices. *Cognitive Science*, 5, pp 121-152.

Clement, J. (1989) Learning via model construction and criticism. In G. Glover, R. Ronning and C. Reynolds (Eds). *Handbook of creativity, assessment, theory and research.* (New York, NY: Plenum).

Cohen, A. (1994) The effect of individual work on collaborative student activity in a CSILE classrrom. Paper presented in *Computer Supported Collaboration for Scientific Inquiry; Bridging science learning closer to the scientific practice*. Annual meeting of AERA, New Orleans, April 1994.

Constantino-Conzalez, M.A. and Suthers, D. (2001) Coaching Collaboration by Comparing Solutions and Trackning Participation, in *Proc. of European Conference on CSCL,* Maastricht, NL, March 2001. pp. 173-180.

Devin-Sheehan, L., Feldman, R. and Allen, V. (1976) Research on children tutoring children. *A critical review. Review of Educational Research. 46*(39), pp 355-385.

Dimitracopoulou, A. and Dumas-Carré, A. (1996) Pour une vision plus elaborée du processus explicatif dans les environnements d'apprentissage en Physique. *EXPLICATION'96*, Sophia Antipolis (ed.) INRIA (Institut National de Recherche en Informatique et en Automatique), France, (pp. 323-342).

Dimitracopoulou, A., Vosniadou, S. and Ioannides, C. (1997) Exploring and Modeling the real world through designed environments for young children, In *7th European Conference for Research on Learning and Instruction, EARLI*, August 26-30, 1997, Athens.

Dumas-Carré, A. and Caillot, M. (1989) Cognitive aids for problem solving. Paper presented at the *Annual conference of American Research Association*, San Francisco.

Guzdial, M. (1997) Information ecology of collaborations in educational settings: Influence of tool. In proceedings of the *2nd International Conference on Computer Supported collaborative learning CSCL.*

Halloun, I. (1996) Why all the excitement about modelling instruction? *Summer Meeting of the American Association of Physics Teachers*, Maryland.

Halloun, I. (2002) Model-laden inquiry for effective physics instruction. In *Themes,* Vol. 1, No4, pp.339-356.

Harasim, L. (1997) Interacting in hyperspace: developing collaborative learning environments on the WWW. Paper presented at *the Workshop for improving economic management training.* Monaco.

Hatano, G. and Inagaki, K. (1992) Desituating cognition through the construction of conceptual knowledge. In P. Light and G. Butterworth. (eds). *Context and cognition. Ways of knowing and learning.* (pp. 115-133). (New York: Harvester).

Hestenes, D. (1996) Modeling Methodology for Physics Teachers, in *Proceedings of the International Conference on Undergraduate Physics Education,* College Park, August 1996.

Huitt, W. (1992) Problem solving and decision making: Consideration of individual differences the Myers-Briggs Type Indicator, *Journal of Psychological Type*, 24, p.33-44.

Hutchins, E. (1995) *Cognition in the wild.* (Cambridge, MA: The MIT Press).

Jackson, S., Krajcik, J. and Soloway, E. (1998) The Design of Guided Learner-Adaptable Scaffolding in Interactive Learning Environments. *Conference proceedings on Human factors in computing system.* pp: 187-194.

Jermann, P. (2002) Task and Interaction regulation in controlling a traffic simulation, In G. Stahl (ed). Proceedings *of Computer Support for Collaborative Learning, CSCL 2002*, congress, Colorado, January 7-11 2002.

Joolingen, W.R., and de Jong, T. (in press) SimQuest, authoring educational simulations. In T. Murray & S. Blessing & A. Ainsworth (Eds). *Authoring tools for advanced technology learning environments: Towards cost effective adaptive, interactive and intelligent educational sowftare.* (Dordrecht: Klauwer).

Kyza E., Golan R., Reiser B. and Edelson D. (2002) Reflective Inquiry: Enabling group self regulation in inquiry based sciences using the progresss portofolio tools. In G. Stahl (ed). Proceedings of *Computer Support for Collaborative Learning, CSCL 2002,* congress, Colorado, January 7-11, 2002.

Laborde, C. and Vergnaud, G. (1994) L'apprentissage et l'enseignement des mathematiques. *In Apprentissages et didactiques ou en est-on?* (Coord. G.Vergnaud, Serie Education: former organiser pour enseigner), (Paris: Hachette Education).

Martinand, J.-L. (1992) Presentation in *Enseignement et Apprentissages de la Modelisation en Science.* (Paris: INRP).

Mellar H., Bliss J., Boohan, R., Ogborn, J., Tompsett, (Eds) (1994) *Learning with Artificial Worlds: Computer Based Modelling in the Curriculum.* (London The Falmer Press).

Miyake, N. (1986) Constructive interaction and the iterative process of understanding. *Cognitive Science,* 10, pp 151-177.

Nickerson, R.S. (1993) On the distribution of cognition: Some reflections. In G. Salomon (Ed.) *Distributed cognitions Psychological and educational considerations*. (Cambridge: Cambridge University Press).

Noel, B. (1997) *La Metacognition,* Serie Pedagogies en developpement, De Boeck Universite.

Norman, D. (1993) *Things that make us smart: defending human attributes in the age of the machine.* (Reading MA: Addison-Wesley).

Ogborn, J. (1994) The Nature of Modelling. In H. Mellar, J. Bliss, R. Boohan, J. Ogborn, C. Tompsett (Eds). *Learning with Artificial Worlds: Computer Based Modelling in the Curriculum,* (London: The Falmer Press).

Palinscar, A.S. and Brown, A.L. (1984) Reciprocal teaching of comprehension fostering and comprehension monitoring activities. *Cognition and Instruction. 1*(2), pp 117-175.

Pea, R.D. (1993) Practices of disctributed intelligence and designs for education. In G. Salomon (Ed*). Distributed cognitions. Psychological and educational considerations*. (Cambridge: Cambridge University Press).

Salomon, G. (1993) *Distributed cognitions Psychological and educational considerations*. (Cambridge: Cambridge University Press).

Schoenfeld, A.H. (1987) What's all the fuss about metacognition? In *Cognitive Science and Mathematics Education*, Ed: Alan Schoenfeld. (London: Laurence Erlbaum Associates).

Schwartz, D.L. (1995) The emergence of abstract representations in dyadic problem solving*. Journal of the Learning Sciences* 4, pp 321-354.

Simmof, S. (1999) Monitoring and Evaluation in Collaborative Learning Environments. In C. Hoadley & J. Rochelle (Eds). *Proc. 3rd Conference on CSCL*, Stanford, December 12-15, 1999.

Slavin, R.E. (1995) *Cooperative Learning.* 2nd edition (Allyn and Bacon).

Slavin, R.E. (1997) Research on cooperative learning and achievement: A quarter century of research. Paper presented at the *Annual meeting of pedagogical psychology*, Frankfurt, Germany.

Soller, A., Wiebe, J. and Lesgold, A. (2002) A machine learning approach to assessing knowledge sharing during collaborative learning activities. In G. Stahl (ed). *Proceedings of Computer Support for Collaborative Learning, CSCL 2002,* congress, Colorado, January 7-11 2002.

Suthers, D. (2002) Collaborative representations: Supporting face to face and online knowledge building discourse. *Workshop proceedings on Inquiry Learning, of Computer Support for Collaborative Learning*, CSCL 2002, congress, Colorado, January 7-11 2002.

Tiberghien, A. (1992) Analysis of interfaces from the point of view of epistemology and didactics. In A. Tiberghien and H Mandl (Eds) *Intelligent Learning Environments and Knowledge Acquisition in Physics.* (Berlin: Springer-Verlag).

Tiberghien, A. (1996) Construction of prototypical situations in teaching the concept of energy. In G. Welford, J. Osborne, and P. Scott (Eds). *Research in Science Education in Europe, (*pp.100-114) (London: Farmer Press).

Van Someren, M.W., Reimann, P., Boshuizen, P.A. and de Jong, T. (1998) *Learning with multiple Representations,* (The Netherlands: Pergamon-Elsevier Sciences)

Vosniadou, S., De Corte, E. and Mandl, H.,(1994) (edited by) *Technology-Based Learning Environments*, Psychological and Educational Foundations, NATO ASI Series, Serie F: pp. 302, (Springer Verlag).

Webb, M. (1993) 'Computer-based modelling in school science' *School Science Review* Vol. 74, No. 269, pp 33-47.

Webb, N. (1989) Peer interaction and learning in small groups. *International Journal of Educational Research* 13, pp 21-40.

Webb, N. (1992) Testing a theoretical model of student interaction and learning in small groups. In R. Hertz Lazarowitz and N. Miller (eds). *Interaction in cooperative groups: The theoretical anatomy of group learning.* (pp 102-119). (NY: Cambridge University Press).

Weil-Barais, A. (1990) Concept learning and experiment modeling in mechanics*. In J. Bliss and A. Weil-Barais (Eds) Special Issue on Psychology of Learning Sciences, European Journal of Psychology of Education*, Vol. V, No 4.

White, B., Shimoda, T., and Frederiksen, J. (1999) Enabling students to construct theories of collaborative inquiry and reflective learning: Computer support for metacognitive development. *International Journal of Artificial Intelligence in Education*, *10*(2), 151-182.

Wojahn, P.G., Neuwirth, C.M. and Bullock, B. (1998) Effects of interfaces for annotation on communication in a collaborative task. In proceedings of the *Conference on Human Factors in Computing Systems* (CHI '98). (pp. 456-463). (Los Angeles: ACM Press).

PART IV

MODELLINGSPACE DESIGN PRINCIPLES AND SPECIFICATIONS

1. Pedagogical settings of use and use cases

1.1 Brief presentation of main learning objectives

Our aim is that after using the project's materials pupils should be able to: understand and appreciate the nature of science and its concepts, laws and theories; use the process of modelling to understand better 'everyday' or 'lifeworld' situations; and analyse critically the results of constructing and using models.

Research studies in the areas of science education and cognitive psychology have demonstrated that participating in modelling activities engages the pupils who carry it out in a learning process And that this learning process consists of the following stages:

- ⇨ the formulation of the problem – the situation to be modelled;
- ⇨ the initial design of the model;
- ⇨ the comparison with other models of the same situation;
- ⇨ the exploration of the model;
- ⇨ the evaluation of the model and its modification.

It is expected and hoped that as a result of this process pupils will reconstruct their representations, build concepts and understand better the relevant scientific theories; that is they will progressively construct for themselves models more similar to the accepted scientific ones.

In this approach, pupils will potentially:

- ⇨ nurture cognitive dexterity for solving complex problems: the system contributes to this through both the specific situations suggested for modelling, and the functionalities of the learning environment, which allow the re-examination and use of previously created models;
- ⇨ develop their creative imagination and critical thought: the system contributes to this through both the provision of multiple representations and its open architecture, which allow pupils to express their ideas and imagination by means of simple manipulations of entities;
- ⇨ develop skills for communication and collaboration: the system supports this through its design features which facilitate communication through intranets and the Internet. Pupils therefore have a means through which they can achieve an understanding of the meaning of comparison of viewpoints, of collaborative problem solving, of division of labour and group work, and (it is expected that) can thus develop a spirit of collaboration.

The use of modelling in various school subjects, (which in the environment to be developed are mathematics, physics, biology and interdisciplinary subjects, such as environmental education) and the associated decision-making processes, help pupils to:

- ⇨ comprehend the need for the creation of models in science (for interpretation of phenomena, interpretation of data, prediction of results and events);
- ⇨ acquire modelling skills;

⇨ construct or comprehend better the basic concepts of the respective school subjects, as well as the structure of that knowledge.

Based on the above, ModellingSpace is designed so as:

1. to allow and support modelling on the basis of analysis of problems and situations in terms of entities and objects, which have properties and are linked through relationships;

2. to allow and support expression by means of visualization of the entities and their properties, as well as of the relationships or rules that affect them;

3. not to require use of formal mathematics, as it is the case in classic modelling systems, but to use a simplified and synthetic form of different categories of systems. This allows the study and creation of models for a wide range of situations, by pupils of various ages.

4. to support various appropriate symbolic graphic representations, which can be used as cognitive tools and learning aids;

5. to support collaborative activities amongst students and amongst teachers through local intranets, as well as through the Internet;

6. to support the teacher in the delivery of the curriculum in existing classroom conditions, as well as in conditions which favour innovation, and thus contribute to the improvement of teaching practices and of curricula design;

7. to support the creation and development of a community of learners around the common subjects of models and modelling, which will include both pupils and teachers;

The ModellingSpace project will focus on students aged from 10 to 16 years old (see Table 1.1)

Table 1.1: Target group

Age	Education level
10-11 years old	End of primary school
12-15 years old	High school
15-16 years old	1st year senior high school

Concerning subject matters, the project will develop learning modelling activities for a range of subject matters such as physics, mathematics, chemistry, biology, etc.. Table 1.2 presents in an indicative way the type of models that could be used in different subject matters.

Table 1.2: Subject matters and categories of models

Subject matters	Models categories
Physics	Qualitative (concept maps), semi-quantitative, quantitative models
Mathematics	Semi-quantitative, quantitative models
Chemistry	Qualitative (concept maps), semi-quantitative
Biology	Qualitative (concept maps), Qualitative with logical operators, semi-quantitative (corresponding to unknown algebraic models)
Environmental education	Qualitative (concept maps), semi-quantitative
Informatics and decision making situations	Qualitative (concept maps), Qualitative models with logical operators

Detailed presentation of subject matters and corresponding examples of themes for study will be included in the DO5 deliverable (‘Scenarios and Activities’).

1.2 Scenarios of use

1.2.1 General settings

The MODELLINGSPACE environment, taking into account the population that it is concerned with in the frame of the current project, could be used in *a school context in the following general cases*:

1. In a typical class setting with or without local network facilities. Students will work by collaborating face to face (in small groups of 2 or 3 students) on a simple or more complex problem to model.
2. In class but with global network facilities and exchanges: The network of schools and a server supporting activities could provide to students and teachers a wider spectrum of possibilities: Exchange ideas on problems, exchange ideas on solutions and reports, locate peers for a specific problem, or classes that work on the same or similar kind of problems, etc.
3. In class with collaboration through networks at a school, national, or European level: it concerns not exchanges but collaborations with a specific aim to investigate one or more specific study themes.
4. From the users’ (students’ or teachers’) home as supplementary to the work done in school class: Given the technological advancement in networks and bandwidths, as well as in services and costs which will be seriously increased, we consider that in the few next years it should be possible for students to work via network from their home, in order to continue a study theme started at school. In this way students will be able to work collaboratively with their school friends and groups in order to prepare a school project, or to advance or finish their homework or school project. Additionally, they would have the time and opportunity to research the material and the experiences of other students or groups in other classes, other schools or other countries.

We distinguish two main key users: a) teachers; b) students. Other possible users are administrators, technicians and teachers-researchers.

In the next sections we will present at a general level the use cases for: a) modelling activities in local networks; b) collaborative sessions through networks.

1.2.2 Use of ModellingSpace in class

1.2.2.1 Students’ use cases

<u>Use cases in class and eventually via local networks</u>

Students could:

1. Study a problem and Create a model
2. Explore verify and modify a given – created model

Moreover they could:

3. Conceive a situation and propose to other groups to solve it, (eventual use of one entities editor with the help of their teacher)
4. Study a detailed example of a modelling process from the report presented by another student.

Lets take an example of a modelling process, concerning a simple kind of problems. A possible sequence of events (described at a high level of abstraction) could be the following one:

A group of students collaborate face to face on a PC in the class or school laboratory. Students have to open a study theme that exists, or to create another one. Students have to analyse the situation described in the problem statement, and creating an initial mental representation to decide on what are the main factors that influence the phenomenon. They have to choose the kind of variables with which to represent these factors. That is, they will have to decide whether to choose entities such as real objects, indicating the properties that are relevant to the problem - e.g. not the colour of a moving car, but its mass - or to choose abstract variables, such as scientific concepts, considering the studied system as a material point. The entities or the abstract variables will have to be found from the library of entities (opening one or more of the available folders).

They have to indicate relations between two variables. In order to define the relations they have to try to find in the library of relations an expression mode for this relation that is the closest to the existing one. (So, if they know and can use an exact algebraic relation, they will define it; otherwise they will choose one of the semi-quantitative relations. If they can infer only a literal relation, they will choose a qualitative one).

During this process, students are invited to take notes of this initial analysis and of their assumptions, in the structured notebook.

After indicating the relations or before finishing their indication, they could execute the model (by 'running' it if it is not a concept map), in order to examine the simulation that the system produces (given the provided relations). In order to interpret the simulations and the appropriateness of this partial model, it could be useful either to check the values of the variables, or to run and examine the representations of the data that the model produces. So they can indicate the variables, whose values they wish to examine the co-variation, asking for the table of values, the typical x-y graph, or any other kind of alternative representation of data such as a bar-chart or a pie-chart. The execution of the model and the graphs' observation could lead them to re-examine and modify the model, changing the relations or the variables or indicating different initial values for the variables, or finally adding new relations.

Finally, they will note down their new hypotheses into the structured notebook, or they will add just a note, by posting a sticky note in the area of the model editor, so as to remember the reason why they have chosen this specific relation.

In the investigation phase students could be scaffolded by special questions or prompts written in the paper or electronic worksheet of students, related to this specific study theme. The teacher that supervises the students' work via the group supervising utility, or via direct contact with the group, could help them by appropriate hints or questions.

The students will predict the modified model behaviour, and will execute it. In order to interpret what happens and to validate the appropriateness of the model, the students will have to run the model again, and ask to see the graphs.

They may think to 'take pictures/images' of this data representation (e.g. an image of the x-y graph, an image of the table of values) in order to be able to explain later why they have validated or rejected this model. These images could be copied in the Report, in order to comment on them later. They can also 'take images' from the model in order to comment on or explain, if later it appears that it is a non-appropriate model.

If the available time is over, students will have to save the model (and automatically, notes will be saved also).

At the end of this session, teacher could see the models and the notes of each group of students, as well as the history of the model creation process, using the analysis tools.

These products will eventually lead him/her to give appropriate advices to the whole class.

At the beginning of the next session, students can open the model (and directly the study themes, as well as all their notes will be automatically opened). They will have to continue the process of modelling until they reach a sufficient for them solution or model.

The modelling process could not be considered as finished until the students prepare and draw up a written report on the proposed model, as well as on the modelling process, including comments on their intermediary attempts. For this last purpose, students could use the appropriate for them analysis tools in order to remember the process followed.

The teacher, who knows the final models of students (via monitoring them) as well as their history (via the structured analysis of the models' creation process) could coordinate the presentation of the reports of some groups of students. These presentations could result in a more general discussion about the modelling process, or about students' conceptual assumptions concerning the specific problem. As a result of these the teacher may invite some of the groups to improve their models.

<u>Categories of created models, according to their complexity: examples</u>

Students could analyse a study theme, and produce a model that could be considered as:

1. A simple model: For instance, they could study an object moving on an inclined plan. This would require one or two sessions, for young students of 13 years old, using only semi-quantitative relations.
2. A complex model: A complex model that incorporates many variables (e.g. in environmental education). In this case, a group of the class will model a part of the problem, while another group will model another part. At the end, they will collaborate in order to produce a common model.
3. Simple models that will need two or more sessions, if they have to support the transition from semi-quantitative models to quantitative ones.
4. A complex model with different levels of sub-models: For instance, problems in physics incorporating different "languages": (e.g. a model in terms of energy and a model in terms of kinematic equations interconnected).

Some problems can be more traditional ones focused on concepts and representations and others more open, focused on the modelling process, or they may be interdisciplinary ones (innovative but related to the current curricula).

<u>Some typical Kinds of models and uses</u>

Students could work on the following:

- Qualitative models of concept maps so as to structure the concepts' relations during final synthesis at the end of a long session, or during initial brainstorming in order to analyse a complex problem and write down the factors that influence it and their relations.
- Semi-quantitative models with 'what influences what' relations.
- Semi-quantitative models for young students.
- Semi-quantitative models that guide the identification and use of appropriate quantitative models
- Quantitative models directly.
- Decision making models with logical operators.
- Mixed models (with different sub-models of different abstraction).

- Mixed models (a set of interconnected models that represent different phases of a phenomenon).

Some sessions could be long: For instance these that support the transitions from semi-quantitative to quantitative reasoning, may require that pupils:

- Study two or three different but internally similar problems.
- Create two or three models of this kind.
- Think about similarities between these models.
- Work on to find out a common entity (more abstract one) and more general variables that could express all of these models/situations.
- Continue and conceive models with more generic entities and so variables.
- Work on new relations, quantitative ones, trying different formal mathematical equations in order find the more appropriate one for a specific case.

1.2.2.2 Teachers use cases

<u>Teachers' main attributed functions:</u>

Teachers in their typical class with groups of students working on their PC connected to a local network, could:

- Propose to students to model an existing previously prepared study theme.
- Propose that they conceive new problems and situations to model.

  -This implies that the "study theme" tool is open, allowing the creation of multimedia presentations of new problems.

  -Teacher may need to support these new problems by creating and inserting new entities (it implies that an editor of entities exist, sufficiently simple in its use), or by searching the public repository of all schools in order to find any appropriate material.

<u>Teachers' roles during learning activities management:</u>

- To propose and readjust with students the main study theme.
- Adjust the folders of entities.
- Adjust files of models created by students using an appropriate file manager system.
- Supervise the work of each group from his/her workstation.
- Ask for the structured history of their interaction in order to see easily, alternative entities and relations that they have used, and identify in few minutes students' difficulties
- To support students in their questions.
- To go around the groups during groupwork in order to give advice according to their questions.
- To have an overview of all their group progress (supervise models state).
- To incite students to present their work in class.
- To manage discussion during groupwork presentations (in all the cases, students present their work to the whole class).
- To readjust the set of problems in the following sessions.
- To invite them to find relevant material in the public repository.

1.2.3 Use cases in collaborative sessions via networks

1.2.3.1 Students' use cases

Behind the classical modes of face to face collaboration that typically is practiced in the classes, there is the potential to exploit and take profit of collaboration through networks.

Collaboration depends on: The task (kind, complexity, etc.), the members, the roles assigned (or not), the exact setting, the age, the available time, the available tools. The specific collaborative activities that ModellingSpace project will implement in schools, with each age group, will be defined in later reports. However it is useful, for design and development purposes, to specify the collaborative modes that appear to us to have high learning potential and not to be difficult to implement in actual school conditions.

Concerning the number of collaborating partners we must distinguish two cases:

- Two main collaborating students: In problem solving activities and specially in a modelling activity, it seems more appropriate to promote collaboration between two main partners (two students). A third member of collaborative group could be considered the teacher, acting mainly as facilitator, allowing students access to the shared working space and communicating with them using the dialogue tools.
- More collaborating partners: could work together in situations where they have to negotiate the solution or the modelling of a complex problem, in order for instance to divide/split a complex problem to smaller parts, distributing the work in different subgroups.

Concerning the collaborative settings:

We can distinguish some learning settings that offer an added value to the actual learning modes (see Table 1.3):

"**Negotiation of differences":** The case when students negotiate their differences concerning the modelling approach to be used on the same "problem". The students have already worked on the problem in their private workspace, and have then made their solutions public. The teacher recognizes the 'interesting' significant differences of the various solutions, and incites students to present and/or argue their solutions.

Two approaches will be explored:

a) each group of students or individual, present to each other group their solutions (their models). The students of the two groups collaborate in order to argue different parts of their solutions and convince the others, or in order to conceive/create a third common solution.

b) the two groups of students or two individuals construct again a common solution, by scratch or by using parts of their existing solutions.

It is the most interesting case to work in a synchronous collaborative mode.

"**Cooperative modelling process**": The case when the students cooperate working on partial models of a complex "problem", and then they produce the global model of the "problem" (e.g. a concept map of a complex environmental situation). This use case may involve an initial session of synchronous collaboration, that will result in the initial analysis of the problem and the repartition of the whole problem to sub-parts, intermediary sessions of asynchronous collaboration, and a final session of synchronous collaboration that will focus on the integration of partial models into the whole model.

"**Collaborative-common modelling process**": The same collaborative scheme may include the case when the students start on-line to create a model of a "problem" from scratch. This situation is interesting and manageable in classroom conditions, when it involves models that do not incorporate many entities and relations, but a small number of them, which are conceptually rich.

“**Student-student tutoring - Peer Tutoring**”: The case when a student is engaged in a modelling process and another student assists him, making suggestions and giving advices. The pair of students is usually arranged or suggested by a teacher, and can be composed of students of the same age as well as of different ages - a learning situation that can be mutually fruitful for the participants. The student-tutor observes the actions of the student tutee, and gives advices, on actions, observations and meanings. It requires a synchronous collaboration mode.

“**Apprenticeship learning mode”:** The case when a well-experienced student is engaged in a modelling process in public and another student observes this process. The advanced student must be able to answer the observer's questions, as well as to justify or explain his/her answers. It requires a synchronous collaboration mode.

“**Learning from examples”:** The case when a student wants to study already published and commented models of a "problem", produced by other students and stored in the common repository of the ModellingSpace server. The student can study an example by one or more of the following modes: a) The student sees the model of the "problem", produces explanatory notes and then compares his notes with these of the model's creator; b) The student can see the report on the modelling process prepared by the model’s authors and/or its elaborated history in a diagrammatic mode. In both modes, the student can communicate with the model's creator without a teacher's ‘presence’ being required. It requires an asynchronous collaboration mode.

Table1.3: Main collaborative modes

Collaborative modes via networks	**Main Kind of synchronous/ asynchronous communication/ collaboration**	**Local/ global network**	**Qualities/ reasons of collaboration**	**Teacher’s role**
Negotiation of differences	Synchronous	Local/Global		After session analysis During session in order to coach, if necessary
Cooperative modelling process	Phases of synchronous, asynchronous, synchronous	Local/Global	Management of a complex problem into a simpler one	During synchronous session in order to coach. After session analysis
Collaborative-common modelling process	Synchronous in simple problems. Mixed of synchronous and asynchronous (in complex problems)	Local/Global		During synchronous session in order to coach. After session analysis
Peer Tutoring	Synchronous	Global		Not necessary/ After session
Apprenticeship learning mode”:	Synchronous	Global	Explanation	Not necessary/ After session
Learning from examples	Asynchronous	Global	Self-explanation	None

It is to be noted that the three last modes are appropriate specially during the initial period using the learning environment, or during a first period that they work on a specific kind of models. The existence of a ‘learning community’ allows students to ask freely different students in different classes or schools for collaborations. Peer tutoring and apprenticeship learning mode is to be applied in cases where collaborators already know each other and have established good collaborative and social behaviour.

Concerning synchronous versus asynchronous collaboration

In typical school conditions, sessions of synchronous collaboration can only take place in short periods of time (15 min-45 min) and specially for:

- exchange of ideas on crucial points; to negotiate meaning and to explain, to clarify specific points.
- negotiating the distribution of a complex task to different groups, or initial brainstorming, so as to analyse the problem and decide on how to proceed

In most of the cases, in order to work on a full modelling cycle, a mixed approach of both synchronous and asynchronous collaboration seems to be required.

Concerning the Location and Time

Students can collaborate in a flexible mode, using different networks, in different locations and time zones:

- Work via local network in a class where in some cases synchronous collaboration could be fruitful, specially for diagnosis purposes, when rich conceptual knowledge is implicated in problem solving. Investigations in classes have shown the significance of this setting both for learners and teachers (see research report in Annexe D), specially for diagnosis purposes (teachers) and for conceptual awareness (students).
- Work via global network, in order to collaborate with:
  - ⇨ students of another class in the same school;
  - ⇨ students of other schools at a national level;
  - ⇨ students in another country at a European level.

It is to be noted that national or European collaborations and specially in a synchronous mode are more difficult to be arranged. However, after having worked collaboratively at a school level, it is fruitful to proceed to a national or European level collaboration, on specific and well chosen themes.

- Free use at home (with colleagues from school, or in other national schools, or European countries): Work at home could be seen as supplementary to work started in the school. Imagine that some students start to work using the local network, and then continue to work at home. Students work collaboratively with their friends and groups from school in order to advance or finish their homework or their school project. They can continue to work collaboratively with a friend in a short session of synchronous collaboration, something already expressed by students of our trial classes (see Annexe D).

  Additionally, they could have the time and the opportunity to research the material and the experiences of other students or groups in other classes, schools or countries, or to work on their own studied problems and modelling reports.

1.2.3.2 Teachers' role

We could consider two modes of teacher-student interaction, using the network:

- "**Synchronous Coaching**": the teacher observes the student's modelling process or the students' collaboration within a workgroup and intervenes to give advices, make suggestions, etc.. In the first stages of collaborative activities, teachers are expected to spend time interacting with small groups encouraging and supporting the pupils, while in the more advanced stages a group of pupils is expected to work more and more independently of the teacher. Thus, the teacher's role moves progressively from one of the group leader, to one of facilitator, where s/he provides scaffolding for the discussion but does not take an active role, and finally, to one of the observer. But observation of several groups working in a synchronous mode, taking place in parallel, is not easy (see previous chapters). In order to allow a teacher to detect when particular children have difficulties, to identify skills which seem to be generally weak across a group of children and also to be able to suggest when a group of children are not working well together, the teacher has to use the appropriate supervision and analysis tools: (a) supervision tools; (b) analysis tools based on OCAF framework.

- "**Asynchronous coaching**": It is based on after session analysis of students' interactions independently whether these sessions have taken place via synchronous or asynchronous collaboration. Teachers, must have in their disposal (a) very rich and detailed data from 'easy-to-read' logfiles, in order to diagnose specific conceptual difficulties; (b) the structured history of collaborative problem solving; (c) a further analysis based on queries.

2. Design principles of MODELLINGSPACE

In the following paragraphs the main design principles of the technology based learning environment that will be developed in the frame of ModellingSpace project are presented briefly.

The main design principles of Modelling Space are related to:

1. Modelling
2. Reasoning support and learning process support
3. Collaboration support
4. Learning Community support
5. Students' awareness and metacognition support
6. The support of teaching process
7. Users' interfaces design and usability
8. Technological approach

2.1 Principles Related to Modelling

1. Modelling should be based on analysis of problems and phenomena to *entities (such as objects or abstract notions as concepts)* to *properties* of entities (simple properties of objects or scientific variables), and *relations* between these properties. This approach is being chosen because it is appropriate for children's reasoning allowing them to express and test their existing cognitive mental models. In parallel, the system should allow reasoning and a model's expression in scientific terms.
2. For a technological environment being really appropriate for young students it is important to gradually support learning progress, as well as knowledge and skills development starting from the existing ones. The environment will focus *in allowing and supporting qualitative and semi-quantitative reasoning*, which is closer to the existing cognitive resources of young students.
3. It should *fully support the transition from the semi-intuitive ideas of students to the scientific ones*. For this purpose, a whole range of basic models will be given to the students' disposal, such as entities that could represent realworld objects with proto-variables and real world simulations, to more abstract ones with modelled objects (eg. modelled as material points or bodies), characterized by scientific variables with literal names or symbols and abstracts simulations.
4. It should incorporate a simplified as well as a synthetic form of different independent modelling system categories: *dynamic quantitative* (algebraic) modelling systems; *semi-quantitative* modelling systems; *qualitative modelling through concept maps;* and *qualitative modelling through logical operators;* so that it allows the study and the creation of models for a wide spectrum of problems and phenomena. These categories of models are able to support procedures and modelling mechanisms that derive from *different subject matters* (physics, mathematics, biology, chemistry, environmental education, computer science) and thus permit working *in an interdisciplinary mode*.

2.2 Principles related to the support of reasoning and learning process

1. The expression through the ***greatest and most appropriate visualization*** is supported. The application of this principle concerns both the entities as well as their properties or variables and the relations that govern them or impinge upon them. Even with abstract ideas, concepts, laws or sentences can be visualized and this can be done in an unequivocal way. Appropriate visualization constitutes a crucial point for the support of the development of reasoning in children and more specifically the support of the transition from reasoning with objects, to reasoning with abstract concepts.
2. ***The combination of modelling tools with real world simulations*** (not abstract ones): The simulations that are being produced from most of the existing modelling systems are merely abstract. It is important to have the possibility to test and validate models through simulations that represent the phenomenon itself in an obvious visual way.
3. The **structural elements** of modelling, the ontology and the structure of the models have to be appropriate for and adaptable to the cognitive level of the students (different ages, cognitive resources and demands). They also have to be compatible with the epistemology of the different disciplines.
4. ***The incorporation of alternative and multiple forms of representations***: To allow different or even **multiple forms of representation** of both the models (entities, their attributes and their interrelations) as well as of the different kind of data produced by models. The students' ability in making and using models depends on the representational tools which are disposable at their command. The multiple representations provide cognitive assistance for reasoning and consequently for learning. Consequently, as regards, to graphical representations, a broad spectrum of representations has to be available, appropriate for the students and the situations that are to be modelled.

2.3 Principles related to collaboration support

1. In ModellingSpace we will take advantage from the positive learning potential of all kinds of collaborative settings, related to time and space dimensions: ***face to face collaboration, synchronous and asynchronous collaboration*****,** through local and wide networks, due to cognitive and social order reasons.
2. Special attention is given to provide appropriate means during synchronous interaction, in order to ***coordinate action*** in a flexible way, through the application of an action key approach to the regulation of action, ownership protocols and functions able to support the workspace awareness.
3. ***Multiple, flexible and linked modes of dialogue*** during interaction are of great importance in collaborative modelling and problem solving in rich conceptual domains: Chats, structured interfaces with sentence openers as well as sticky notes are provided during synchronous interaction, while during asynchronous interaction the main tools for text annotation will be mainly used, enriched with the function of 'keep track' of each participant contributions.

2.4 Principles related to students' awareness, self-regulation and metacognition support

1. In order to promote ***metaconceptual awareness***, we need to provide to students multiple and flexible tools, so as to facilitate them to write down their thoughts during the different instances of a modelling process, and to give them the possibility to come back and think upon their thoughts and the evolution of their ideas. These tools could be: ***Sticky notes***, a free form to note down something (a short paragraph) in the working area, in order to specify or remind oneself of something; ***Structured notebook***, that could invite students to note their thoughts during each initial analysis, expectation, or observation, and the ***Final Report***, that concerns the whole modelling process, with arguments and data that will accompany the final model.
2. In order to promote ***metacognitive mental*** activities on part or the whole of the modelling process, we need to support students to come back and reflect on their own process and evolution; we need to provide a support to their memory. In other words there is need for appropriate tools that offer traces of the group's previous activity and ***support the group-memory*** (e.g. the memory of the face to face collaborative group of students).
3. In order to support ***reflective inquiry by scaffolding,*** it is important to create and customize templates that address the specific goals and sub-goals during a modelling process. The final report as well as the notebook could provide to students appropriate ***prompts*** (depending on their age and the task category) that may scaffold their activity. The scaffolds will be explicitly conceived to guide students to acquire general inquiry skills.

2.5 Principles related to the Learning Community support

1. Apart from some typical tools such as threaded discussions and whiteboards for announces, the attention is focused on appropriate ***open and interoperable repositories***, provided not only for storing and accessing intermediary or final products but also to store histories and analysis elements, as well as other additional external materials in various common formats. The whole approach tries to support sustainability and reusability of the work done.
2. ***Awareness of new actions and events*** from the community, as well as awareness of new materials added or any changes related to the repository are supported.
3. ***Flexibility of use from different places*** at different times must be supported, while group and community memory will be supported by ***logfiles of the global community level*** actions.

2.6 Principles related to the support of the teaching process

1. The success of collaborative environments based on argumentation and explanation ultimately depends on the possibility of students and teachers to evolve new practices from existing ones. During this process both teachers and students must be gradually supported. The main assumption is that difficulties in applying collaborative learning environments in real school conditions are due in a great degree to the lack of ***appropriate tools supporting teachers***.
2. It is considered as a minimum requirement to provide ***tools*** that allow ***supervision*** of students' screens.

3. Important for teachers that want to diagnose students' process and their conceptual or strategic difficulties, are tools that based on the ***logfiles*** make them more easy ***readable***, providing additionally link between the dialogues history and the state of the common workspace, ***reconstructing the state of the shared working space***, in a chronological order.
4. Significant support should be provided by an ***automated analysis of students' interaction***, presented in a diagrammatic form based on an object oriented collaborative problem solving analysis framework. This analysis, being more detailed and essential when a student uses the structured dialogue interface, is based on a unified analysis of actions and dialogues.
5. Queries that could exploit furthermore the previous analysis, provide to teachers more ***elaborated indicators*** on the quality of the model produced as well as on the quality of the collaborative process.

2.7 About the design of the interface and its usability

1. To allow ***definite distinction among the levels of work and reasoning***, that is between the empirical (cases of the real world) model development (and their constituents) and the level of control of the appropriateness of the modelling process (execution of the model), so that it promotes understanding without fostering confusion.
2. To allow the ***direct manipulation of the interface*** offering an appropriate ***combination of choices*** with icons and verbal choices. The importance of the choices has to be jointly recognized and tends to the 'decrease of the conceptual distance'. The application of this principle is absolutely crucial for the structural elements of the models, like the entities and the relationships.
3. The materialization of the interface and the ergonomics of it should be appropriate for the age of the children that we aim at, with simultaneous satisfaction of two criteria: *"**simplicity**"* and "***minimization of the distance of execution***" (that is the distance between the aims and the intentions of the student and the chain of actions required for the execution).
4. ***Freedom and Constraint:*** The right degree of freedom and constraint can lead students to concentrate on the most fundamental aspects of the task (Baker *et al.,* 2001).
5. ***Negotiable and flexible environment:*** The system and its implementation has to be flexible enough to be adapted to various national cultures, different educational contexts and ages of students. Some functions of the system can be tailored to specific pedagogical purposes. For instance, sentence openers of structured dialogue interface, should be fully editable by teachers or even students. Similarly, the names of entities as well as their variables should be fully editable by users. Usability of a learning system does not depend and is not defined by examining the interaction of learners in a laboratory; it depends on the context and the conditions of use. Designers and developers should not confuse their intentions in developing tools for students or users with the actual way that these tools will be appropriated. In all the cases that it is possible, it could be preferable to build in the possibilities that students and teachers can adapt and negotiate tools to their perceived and progressive needs, during an extended period of time.

2.8 Principles related to the Technological Approach

1. The realisation and the functionality of the system must be ***component object-oriented***. This requirement constitutes a technological demand (that fulfils needs of

reusability of the components, expandability of the whole, etc.) and as regards the functionality, it suggests a cognitive demand as it is connected with principles of learning psychology (Ogborn, 1997).

2. It should be ***open*** to the student-user as well as the teacher-user, in significant aspects. This demand refers to both the study themes, which can be proposed, as well as the structural elements of the models from all categories, that is the entities and the attributes of them. It refers also to the **Repository**.
3. The ModellingSpace environment is intented to be distributed widely across schools and educational institutions. Thus, it is necessary that the software has a sufficient level of ***technical usability***: it is easy to install and to sufficiently run on local servers without problems, given that most schools have not an advanced technological infrastructure nor a technical support.
4. It is desirable to support ***sustainability*** and ***reusability*** of the work done, offering a real ***interoperable*** system.

3. From Design Principles to Initial Specifications of ModellingSpace Environment

It is expected that all the specifications will not be equally important. Some requirements may result to be essential, while others may result to be desirable. Some requirements will be relevant from the prototype stage and others will be applied only to the final application.

Each specification shall be annotated to make these differences in relative importance clear and explicit. On the one hand, this will help developers to give more careful consideration to each specification, by clarifying hidden assumptions they may have. On the other hand, it will help developers to make correct design decisions, and devote appropriate levels of effort to the different parts.

Requirement annotation will address at least:

- *Degree of necessity*
- *Time necessity*
- *Stability necessity*

Degree of necessity

Requirements will be distinguished as: Mandatory;, Desirable; and Optional.

Time necessity

Requirements will be distinguished as: In first prototype; In second prototype; In the final application; Open to development.

Stability necessity

Requirements will be distinguished as: Stable; Flexible; and Unstable.

Default

Where not annotated, a specification will be regarded as: Optional; In the final application; and Stable.

3.1 Main Areas of the learning environment

The basic constituents of MODELLINGSPACE (thereafter MS) are shown in figure 3.1. The main areas are:

- 'Design -Testing of Models' area, for which related components are:
  - ⇨ Modelling-design tools: constituted by entities, relations, and the models' design toolbox.
  - ⇨ Representation tools.
  - ⇨ Annotation tools.
  - ⇨ The 'Entities editor' for the definition of new entities.
- 'Themes of Study' area: This provides the user with the themes to study, or allows him/her to prepare a new theme with appropriate multimedia material.

The spaces and tools that concern the network based learning community, as well as the tools enabling and supporting collaboration through network will be presented in the next section (3.2.). The main components are the following:

- Components addressed to the whole learning community, the most important of which are:
    - Repository of materials;
    - Group management tools.

When working in a collaborative mode during problem solving and modelling activities, the following tools will be used:

- Dialogue tools; and
- Coordination of action functions

Finally, in order to support the modelling process and reflection on it we include:

- Tools for supporting teachers ('Teachers' assistance tools')
- Tools for supporting students' meta-analysis ('Students' meta-analysis tools')

In their use of the system, users are assisted by:

- Files' Manager
- Help System

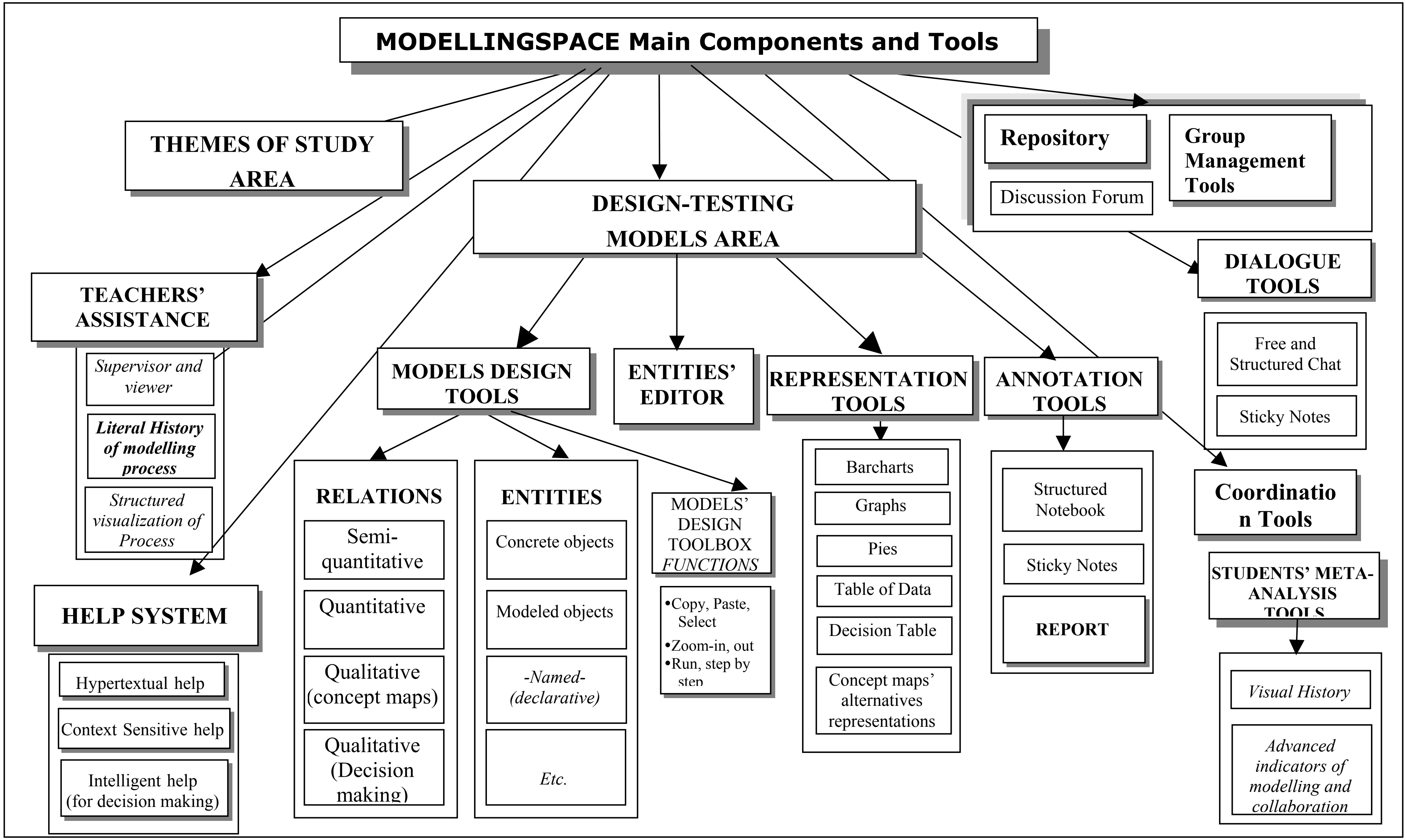


Figure 3.1: Main components and tools of MODELLINGSPACE

3.2 Space for the Creation-Testing of Models

3.2.1 *"Design and Test Models"* Area - Structural elements for creating models

All the requirements concerning the 'Design-Testing Models' area presented hereafter (i.e. in the section 3.2.1) have to be considered as following:

Degree of necessity: ***Mandatory***

Time necessity: ***In first prototype***

Stability necessity: ***Stable***

Except if it is specifically indicated in a different way.

3.2.1.1 Introduction: Models and their structural elements

The purpose of MS is the creation and test of models for the expression and study of phenomena, situations, and problems. The MS as a modelling environment allows to implement and test models.

A model, which is created in MS, can be generally defined as:

«a collection of basic formal entities, or complex entities, that have or commend explicitly predefined properties that can be associated with each other with well defined relations».

The aim of the model is to represent or simulate with satisfactory 'constancy' the intrinsic parts of a certain space of reality.

This definition includes a wide variety of entities/situations, as well as a wide range of different kinds of models. The structural elements of a model in MS are in their general form the entities, the properties of the entities and their relations.

In order to give students means of expression that are appropriate for their cognitive resources and abilities and to support them in the learning process, the following four special model categories have been chosen:

1) Quantitative models: These act in measurable dimensions and algebraic forms and represent the interrelations that correspond to simple or complex mathematical equations. The dynamic mathematical model of a phenomenon presents the evolution of a phenomenon or a system or how discernible situations of it evolve.

- **Structural elements/properties:** The expression of algebraic quantities of models is attained through variables that usually express general scientific concepts and have a wide range of application in a broad range of situations, while they comprise properties or situational features of a large sum of uniform entities. In the variables of the quantitative models a distinction between the independent and the dependent variables is contained.

In MS, the creator of the model will be able to express these variables in two ways:

a) as a general algebraic expression through symbols (without a direct link with a specific situation being shown);

b) as an algebraic expression between properties of specific entities that have been inserted in the active space of models' creation.

- **Structural elements/relations**: The models' creator can express in MS quantitative algebraic models that belong in one of the typical forms of algebraic relations (from y=a*x, to y=ax2+bx+c and from logarithmic to trigonometric relations) that students meet in secondary education.

- **Interpretation of formalism by the system**: The system recognizes the algebraic relations and equations, as well as the numeric values that correspond to the parameters of the equation.
- **Running of the model**: Running the model results in the calculation of a series of numeric values in the table of values, and their graphical representation in forms chosen by the user. In the case where the algebraic formula between the variables is not expressed abstractly (that is using x, y, z etc) but more concretely as relations between the properties of entities, the system can, when running the model, simulate the behaviour of these properties.

2) Semi-quantitative models: While they can be dependent on quantifiable variables, they do not necessarily represent their values. The semi-quantitative models contain the distinction between independent and dependent variables of a model. They can be implemented in situations described by quantitative models. They express reasoning, which regards the world through entities that have properties and not through general concepts and variables like quantitative algebraic models do.

- **Structural elements/entities:** Entities that represent objects or systems, real or imaginary, that are characterised by properties.
- **Structural elements/properties:** These refer to properties of concrete entities that belong to the model. These properties contain the distinction between independent and dependent variables. They can recognise order of magnitude but not values. They can be expressed both as a) properties that can take a certain numerical value; and as b) those that cannot be calculated with accuracy but can be said to have 'low, medium, or high' value.
- **Structural elements/relations:** In semi-quantitative models relations connect properties of entities. The formalism of semi-quantitative relations, as it has been shown in the present phase of analysis, gives the chance to express covariant relations between usually two properties, like the following: «when the one dimension increases the other increases too», «when the one dimension increases the other decreases» etc..
- **Interpretation of formalisms by the system:**
    a. **when they correspond to exact mathematical relations:** The relations that express the variation between two variables, are interpreted internally into 'constituents' of equations. For example: «when one size increases the other increases too» is interpreted as «y=a*x», where *x* is the dependent variable, *y* the independent variable and *a* coefficient, that has by default been given a value by the system or it has been set by the user as the fixed ('locked') value of a third property.
    b. **When they correspond to 'what influence relations':** When they do not correspond to known underlying mathematical relations, it remains to be explored how the system will interpret these relations in order to execute the model.
- **Running the model:** When running a model which consists of semi-quantitative relations, the system initially calculates numerical values using the equations that correspond to the semi-quantitative relations that have been selected by the modeller. Based on these calculations, the corresponding tables of values and graphs for the properties are produced. A processing of the values of the selected properties, in combination with images that correspond to these values, create the simulation of the behaviour of the entity.

3) Decision-making models: The decision-making models implement logical rules (using logical relations in their composition) and result in conclusions and decisions.

- **Structural elements/entities:** Entities that represent objects or systems, real or imaginary that have properties or behavioural conditions.
- **Structural elements/relations:** The relations between the properties of the entities and their values are expressed through logical operands (IF- THEN-OTHERWISE-AND-OR-NO) that determine the formalism of the models in this category.
- **Interpretation of formalism by the system:** The system recognizes the structure of the model, the logical operands and the various structural elements.
- **Running the model:** The model is transformed into an equivalent model which is comparable with the reference model that the teacher has created. The reference model is in some way the target-model for the students. The comparison mechanism checks the correctness of the various proposals of the model, while at the same time runs checks on the occurrence or the non-occurrence of elements from the model (entity, property). That is, it checks whether a wrong property of an entity is chosen; or if a wrong entity is chosen; if the indication of the values of the properties of some entities are not appropriate, and if the logical rules are used well. In all the above cases the system generates a message to the user guiding him towards the correct answer.

  It is to be noted that decision making models are annotated as follows:

  Degree of necessity: ***Desirable***

  Time of necessity: ***In second prototype***

  Stability of necessity: ***Stable***

4) Concept Maps, consist of static graphical representations of a group of factors and/or concepts and their relations. The construction of such a model, helps us to analyse and understand better a complex situation like for example the water cycle, the system of transport in a city, fire in the woods. Concept maps are static models. Static models represent a phenomenon, a situation at a certain moment in time (a forest at the time of fire). To depict the evolution of the situation through time, we will have to construct a new model (a forest in winter after a fire). Such a static model can change, only if our knowledge of the factors or the relations which are being represented changes.

- **Structural elements/entities:** The entities can be, imaginary or real objects, systems or general concepts and ideas, and can be represented with words or icons. The modeller can implement different hierarchical classes of entities.
- **Structural elements/properties:** The entities do not have properties as structural elements.
- **Structural elements/relations:** The modeller can determine the relations between the entities through a non-prescribed formalism, rather through literal type expressions between two or more entities.
- **Interpretation of formalism by the system:** The system can recognize only the different classes of entities (that express a hierarchy), and not their relations.
- **Running the model:** The model is not executable.

3.2.1.2 Entities

Analysis-Definitions

In the real world, when we want to describe, analyse, predict, or explain a situation, that is when we want to model it, we reflect first of all on the entities that participate in it.

Studying for example instances of real life, the entities that usually take part in them are of the following form:

i) Simple objects of the real world: for example car, ball, box, each of which can be the 'moving' object in the study of motion.

ii) A complex object (man, car) or system (river, factory, machine, 'people in a town', 'animal population'), with some of the factors that affect or determine its behaviour.

iii) General more abstract conceptions (e.g. production, consumption).

iv) An object can be considered an entity, which plays a role in a situation (and thus in the respective model), but this same entity can also be the object of study and creation of another self-existent model. An entity of a model can be analysed in one or more supplementary and capable of being connected models.

v) In many cases we work at an abstract level, beyond certain entities/objects that enter into the situation under study. We can therefore consider abstractions-simplifications (models) of simple material bodies: e.g. material point (a range of material bodies can be considered as such: e.g. car, ball, box etc.). In this case an imaginary construct is being used, that can represent the common characteristics (e.g. the akin behaviour with regard to motion) of a broad set of objects.

To sum up, depending on the way we analyse a case under study, the entity can be a physical or imaginary object, simple or complex, viewed under a certain lens and with the help of the appropriate formalism each time.

In the environment of ModellingSpace «each entity entering into a model is considered a 'construct' depicting an object from a group of uniform objects real or imaginary that have meaning in the context of a phenomenon, a system, a process or a speculation. The entity represents in general an abstract concept that describes the common characteristics of a set of uniform objects.»

- The entities in (logical) decision-making models: They express an object or a system. Some entities possess external (or internal) characteristics as properties (e.g. car colour, human name, etc.) and others express situations (e.g. car in motion), behaviours or actions (the decision and the action of a human to cross a street given some conditions, etc.).
- The entities in semi-quantitative models: These can have behaviours or can participate in actions, which are quantifiable or not. The entities involved in semi-quantitative models act as reference, out of which the relations between variables spring. Their properties can refer to surface characteristics of an object or a system, or to concepts/variables that play a role in the situation/phenomenon under study.
- Entities in concept maps: In these models, what counts as property is not open. The user can connect entities/objects not only with abstract concepts and variables, but also with properties expressing complex systems.
- Do quantitative models have properties?: In quantitative models relations are expressed through variables-abstract concepts, which can have or have not immediate reference to entities. An abstract model, such as the mathematic/algebraic model, consists of structural elements that constitute variables in symbolic form, which do not constitute entities. However, they acquire interpretational strength if they are reported as properties of a set of uniform entities.

In conclusion, the entity represents a basic component-structural element of models in MS, only in the case of the semi-quantitative and decision-making models, and optionally in quantitative models. In the case of static models of concept maps, the structural elements can be entities or properties.

Additional specifications regarding the entities

Categories of entities in entities library:

For MS, the following three categories of entities are suggested:

- Objects entities: They refer to real objects and have the following subcategories, in relation to the type of property which characterises them: a) Objects-entities with properties that can take numerical values, b) objects-entities with boolean properties, c) objects-entities that can take boolean values with some probability.
- Entities of literal or figurative definition without properties.
- Concepts/variables: abstract concepts-symbols (e.g. x, y, z, S, t, V, a, etc.).

We should also note that an accurate and direct grouping of the objects-entities either thematically or according to the problem classes they belong to, is maybe required.

Creation of new entities:

- Every entity should be linked to a ‘definition’, which could play the role of an extended literal definition-comment for the entity. This description could be provided at the point of the creation of the entity and should be easily accessible at any time of use of the entity in the models’ design space.
- In the case of the entities which are defined literally and do not contain properties, there should be an option for writing indices (e.g. when writing a chemical combination). Similarly when writing the properties of an entity there should be an option to write the type of index.
- The user should have the possibility to define the size of an entity’s representation. This is necessary in some cases, like in study themes in which there is motion of an object in space.
- A dialogue window is needed for the administration of the entities library (e.g. to help the user insert a new entity in the library, to change its place, etc.).
- Creation of a new «complex entity», that constitutes in itself a “product” (an outcome) of a model. The user will have the possibility to define an entity that would ‘hide’ inside a model that has been previously created.

The entity in the models’ design area:

- The user should be able to open and read easily the description of an entity.
- The default size for the representation/simulation of an entity should be standard by default, but also some entities should be able to have larger sizes. The possibility to have ‘variable size’ is deemed useful at least for those entities that are not concrete or that are defined literally (in an open or closed way) and not by an image. The size of the corresponding properties’ space can have a default size, when this is considered adequate.

Background of a model’s construction:

- The possibility to insert background in the models’ design area should be supported, as it is useful for the entities of some study themes, in order to obtain a visual reference to the entire problem. The size of the picture should be determinable.

3.2.1.3 Properties

Analysis-Definitions

The terms ‘property’, ‘variable’ and ‘concept’ are frequently used as synonyms. Generally, the term ‘property’ is used when we want to make a reference to the ‘entity’ which it describes and of which the aspects it represents. The term ‘variable’ is used in the mathematical sense, while the term ‘concept’ is mentioned to point out the cases where these properties constitute 'scientific concepts' and have been defined as such in a certain scientific field.

Special attention is required, from both the designers of the system as well as the future users (mainly teachers) when attributing the entity properties so that the following three conditions are met. The properties should be:

a) epistemologically 'compatible' with the relevant entities or concepts, or

b) tangible to students, and

c) functional for the construction of models in MS

For example, it is necessary to bear in mind that in physics we discern three different categories of properties:

- Intrinsic properties of an object, like volume, mass (and not weight), electric charge, colour, K constant of spring. For example, variables mass and charge are internal/intrinsic to the object electron, whereas the form and size are internal variables for a solid body. These variables have preset values for a certain body.
- Condition properties or correlation properties, like position, speed, tap supply (with values in physics that depend on time). It should be mentioned that one variable that can be considered as condition property for one model can be deemed internal property of an object for another model. Mass for example is a condition variable for a rocket model, whereas for most bodies it has a constant value. On the whole all properties of an object can be considered condition properties with a constant value.
- Interaction properties, like force, potential energy, etc.. These properties express interactions between two entities (force is a concept that models the interaction between two objects, while the potential energy of an object expresses the interaction between the object of study and the object earth), and as a result they cannot be considered as intrinsic properties of an entity/object or as condition properties of the object. Work, and momentum are interaction variables as well.

  Akin analyses are important to be carried out in the different cognitive subjects that the designers of the system and the teacher users will examine for the selection of themes of study or the creation of new entities.

On top of these, it is necessary to make the distinction between:

- Basic Properties: These are properties that are independently defined (like time, position of a mobile, mass).
- Derived properties: These are conceptualised with the aid of other basic variables (e.g. 'supply' from volume of water and time, speed from position and time). It is a matter of choice for the designer of an entity if it is defined in advance by an algebraic relation at the definition stage of the entity (so it would be thereafter automatically calculated) or if it is left to the user to check and define the dependence of this variable through relations and during the creation of respective models.

The Type of property is another factor that determines which entity properties can be interrelated: Properties with the same type can be interrelated. We distinguish the following two general types of properties:

A) Entity properties that participate in semi-quantitative or quantitative relations.

B) Entity properties that participate in logical relations.

Another distinction which is necessary when creating a model, for all the properties that participate in semi-quantitative or quantitative relations, is this between:

- Independent variables, and
- Dependent variables: These are the variables which we wish to study the behaviour of, or calculate the values.

  Whether one variable is dependent or independent will be expressed with relation arrows that have a certain orientation, and will have consequences when the model is running.

Supplementary specifications concerning properties

Kinds of values of properties: Properties of MS entities should be able to accept the following types of values:

a) integers (also labelled by the system as low, medium, high);

b) ratio;

c) boolean with values (true, false) but also with the possibility of receiving values;

d) nominal, alphanumerical (e.g. date);

e) sets;

f) ordered sets.

Values of properties that participate in semi-quantitative relations:

The user should be able to set an **initial value** in any **variable**. An initial value for a variable (property) should be determined at the time of definition (by default the value is zero). This default value should be able to be reassigned at the time of use of the property of the entity, in the models' design area, when running the model. We should examine whether it is possible to leave a trace near the value slider, which shows that there is a diversified initial value, e.g. a small arrow (e.g. red) with which the user can indicate the initial value of the variable (a negative value, 0, a positive value)

Some properties should have the possibility to receive negative values as well as positive. As a result, in such a case zero could be indicated on the slider.

The values that are registered as constant (by placing the slider and locking the padlock) should be recognizable by the system.

3.2.1.4 Relations and Models

Supplementary specifications concerning relations

a) Semi-quantitative relations

1. The user should be able to express the sum of two properties or relations, i.e relation Y=X+Z where X, Z are either properties (e.g. speed) or relations (e.g. V*t), thus segments of the model.

2. Possibility to express a multiplication relation e.g. Y=X*Z, where X, Z are either properties or relations, thus segments of the model.

3. Possibility to express the derivative of a property in relation to time (i.e., the rate of change). This is important both in physics (e.g. a=dV/dt) and in mathematical scenarios: e.g. tap flow.

4. When developing the system it is necessary to examine carefully the sensitivity of the semi-quantitative relations (at the internal interpretation level of the model), especially in complex models in which we do not know the exact mathematical relations.

5. In an approach that considers semi-quantitative relations as parts of quantitative relations (see respective list in following sections), we should include at least the following relations:

- When an independent variable increases, its dependent increases too (proportionally, i.e. corresponding to equation y=a*x).
- When an independent variable increases, its dependent decreases (inverse proportionally, i.e. corresponding to equation y=a/x).
- When an independent variable increases, its dependent increases more (corresponding to equation $y=a*x^2$).
- When an independent variable increases, its dependent decreases more (corresponding to equation $y=a/x^2$).
- The dependent variable expresses the change of rate of the relevant independent variable (corresponding to the equation of derivative: y=dx/dt)
- Trigonometric relations.

<u>Combination of relations (a kind of Hybrid Models)</u>

6. It should be explored whether when the system expresses models, the relations of «rational operands» [(IF x>a, IF x<b, IF a<x<b) THEN .... *a specific relation*] can be used as control structures.

7. It should also be explored whether it is intentional to allow the linking/connection with abstract variables that typically participate in quantitative relations.

<u>Form of representational relations in the interface</u>

8. Great caution is required so that appropriate <u>visual representations</u> are sought for the relations, so that they do not lead to misinterpretations. Concurrently, appropriate verbal expressions for the relations are required.

10. An appropriate way of visual differentiation between multiplication and summative semi-quantitative relations is required.

11. An appropriate way to express the rate of change (derivative) is to be defined.

b) Quantitative relations

1. Quantitative relations, in general, should allow the expression of a wide range of equations. In secondary school, students meet the following forms of relationships of the general form y=f(x):

- Y=a*x, proportional relation
- Y=a*x+b, linear relation
- $Y=a*x^2+b*x+c$, second order polyonym
- Y=(a*x+b)/ (c*x+b), homographic relation
- $Y=x^n$, power relation
- $Y=e^x$, exponential relation
- Y=logx, logarithmic relation
- Y=sinx, cosx, tanx, trigonometric relations

It should be examined if it is possible to provide the possibility for expression of polynomial relations (*Optional*).

<u>Hybrid relations</u>

2. The necessity of coexistence for relations of rational operands (control structures) along with quantitative relations should also be examined in a unified model.

<u>Interface of expressing quantitative relations</u>

3. To determine an appropriate interface for expression of quantitative relations.

4. It should be possible to link quantitative relations with the appropriate entity properties, typically participating in semi-quantitative relations. This functionality allows the creation of simulations when running the quantitative model, which would go beyond the graphical presentations and the tables of data.

<u>Quantitative relations and measurement units of values</u>

5. In quantitative relations there could be an indication of the measurement units of the variables, eventually without needing to check their compatibility.

c) Qualitative logical relations

The creation of models, where the use of brackets is necessary, is considered desirable not only for familiarising pupils with the combination of logical operands but also in order to allow them to express complex logical sentences.

MS could contain the following logical relations:

a) **Hypothesis : IF** property I1 of entity O1 has the value T1…..

b) **Conclusion: ...Then** property I2 of entity O2 has the value T2.

c) **Alternative Conclusion: ...Otherwise,** property I3 of entity O3 has the value T3.

d) **Coupling: AND** – it links two conditions in one assumption or one conclusion.

e) **Disjunction: OR –** it links two conditions so that one of them is valid in any moment in the assumption or conclusion.

f) **Negation: NO -** it excludes the value of an entity's property at a given time.

MS could support the creation of «complicated» models with the use of operands AND, OR, where the operands can be linked in any degree of complexity, with the use of brackets .

Whereas the existing software models support sentences of the following form:

[(If Entity1 Property1=Value1 AND/OR (IF Entity2 Property2=Value2)] [(THEN Entity 3 Property3=Value3) AND/OR (THEN Entity4 Property4=Value4)]

MS could give the possibility to express models in the following form:

[IF (Entity1 Property1=Value1 AND Entity2 Property2=Value2 OR Entity3 Property3= Value3) THEN (Entity4 Property4=Value4 OR Entity5 Property5=Value5) AND Entity6 Porperty6 =Value6)]

Thus, the syntax of a model where logical relations are used will have the following form:

IF (Entity1 Property1=Value1 AND/OR Entity2 Property2=Value2) THEN (Entity3 Property3=Value3 AND/OR Entity4 Property4=Value4) OTHERWISE (Entity5 Property5 =Value5 AND/OR Entity6 Property6=Value6)

In the above form, the involved theoretical entities 1-6 can correspond to fewer entities at the design level, even to one entity. Same applies to their properties and values.

In a hypothesis or a conclusion more than one condition could be combined, and these could be linked with the use of the logical operands AND, OR.

Every value of a property would be able to accept the negative operand NO.

d) Relations in concept maps

1. We need to re-examine, in the course of the development and pilot use of the software with pupils, if the open entity exemplars for concept maps are sufficient.

2. There is need to provide a usable interface for the relations implicated in concept maps.

Links between different models

In semi-quantitative models (and possibly in other model categories), the possibility to run two or more different sequential models, the entities of which have common variables, is required.

A possible scenario which fulfils this requirement is:

A) The user is able to design in the working space two or more models, and

B) to register in an appropriate way the links between the models (e.g. the exit of the variable of the first, is the entrance to the next).

C) With the choice of running a model the system runs sequentially the one model after the other.

For example: for modelling the motion of a car three different models are needed (model for the accelerating motion – launching; model for normal motion; model for decelerating motion - braking). The user must link the time of the first model with the time of the second, the speed of the first with the speed of the second etc. via for example an appropriate discontinuous double line. In this way the final value of a variable of the first model becomes initial value of the next model.

3.2.2 Tools for helping the design

This section describes the tools that MS will provide for the creation of models. The functions described here are valid for all model categories.

Support of simple operations:

In order to support the user when designing a model the following functionality will be offered:

- Selection
- Deletion
- Zoom: the facility to zoom in and zoom out will also be given.

Image capture (copy / paste functionality)

This functionality must be offered in the model design area, in order to capture the image of a model.

It must be available also for data representations (graph, table of values, etc). It will allow students to copy the image of a particular data representation, and paste it somewhere (for instance in the associated Report) in order to examine it later.

All the above requirements have to be considered as having:

Degree of necessity: ***Mandatory***

Time necessity: ***In first prototype***

Stability necessity: ***Stable***

3.2.3 Alternative representations of data

3.2.3.1 Graphs

A) Graphs of covariance in two dimensions

1. They should allow the expression of measurement dimensions in graphs.

2. The scales of the axes of the graphs should be externally re-determinable (besides a default allocation).

3. Parallel occurrence of graphs from the execution of two or more different relations between the same variables should be allowed. In this way, in the same window two or three graphs could be superimposed.

4. Different variables' graphs should be differentiated by different colour lines.

5. The presentation of graphs of more than two variables in relation to a third variable should be allowed.

B) Graph of barchart type

Bar charts should show the measurement scale in the variation of the numerical values. There should be the possibility to have horizontal or vertical bar charts.

C) Pie chart

It is useful to develop a graphical representation of a 'pie' chart that shows the percentage of participation of factors in relation to their set (this need arises from scenarios of the environmental study as well as from math scenarios that refer to statistic concepts, 'percentages' etc.)

3.2.3.2 Table of Data

It is important to support the open table of data and this in connection with the quantitative and semi-quantitative relations. It is useful:

1. To allow the possibility of presenting the units of measurements of the properties (that participate in semi-quantitative and quantitative relations).

2. To examine the possibility of enrichment of the table of values, with at least a few minimum computer spreadsheet functionalities, like: SUM, Rund, % (requirement that may result from future math and environmental scenarios).

3. To support the synergy of MS's table of values with a spreadsheet software, so that it is possible to import and export values from one to the other.

All the above requirements have to be considered as having:

Degree of necessity: ***Mandatory***

Time necessity: ***In first prototype***

Stability necessity: ***Stable***

3.3 Annotation Tools

Three kinds of annotations tools are specified: The structured notebook, the sticky notes, and the Report.

All three annotations tools must be linked with the model under investigation. This means that when a model is created and saved also the associated notebook, and report will be saved.

All three annotation tools may support 'keep track' functionality, and differentiation between different authors.

3.3.1 Structured notebook

- The "Structured Notebook" allows students to take notes during the modelling process. The specially designed notebook could contain at least three folders, for instance: "analysis of situation", "model design", and "model validation'.
- Its aim is to encourage students to put their thoughts down during the different successive modelling phases. It gives students the possibility to recall these notes later, working on the model, in order to revise them or think about the evolution of their ideas.

- The names of the folders must be fully editable (in order to be adaptable by teachers or even students).
- Concerning the number of folders the notebook must have, it must be amenable to change.

All the above requirements have to be considered as having:

Degree of necessity: ***Mandatory***

Time necessity: ***In first prototype***

Stability necessity: ***Stable***

3.3.2 Sticky notes

- Sticky notes can be posted in the models–design area. They must be resizable. They should be linked to (a) specific positions of the space, or (b) to specific 'objects' that are already in that space, or (c) to other sticky notes.

All the above requirements have to be considered as having:

Degree of necessity: ***Mandatory***

Time of necessity: ***In first prototype***

Stability necessity: ***Stable***

3.3.3 Report

The report is a modeling support tool, aimed to promote reflective modelling activity. It must be flexible enough to be used both from students and teachers, in their respective activities, by allowing them to create and customize templates

Report form

- It will contain structures for organizing information in ways that are meaningful to the students, reasoning through investigation. Users can re-order and label internal pages in any way that make sense to them. They could even link pages together.
- Visual layout of Report pages: This concerns the way the pages are presented; they should allow easy browsing.
- Students can import previous notes and texts, as well as images. Students can merge here their notes (notebook), sticky notes (in model area) or even insert extracts of dialogues and arguments expressed during collaboration. They can even use sticky notes in order to comment on the images inserted (data or models). By using the data capture functionality, students could move between the report working area and the modelling area. When the student is in the modelling area environment they can identify and capture image of the data or parts of the model, which they deem important, and then they can move to the report working area where they can paste them and store them for further elaboration.
- Report Features:
  - ♦ Pages like Folders (like these in Excel, named by students themselves)
  - ♦ Sticky notes (annotations)
  - ♦ Insert text (with different colours, styles, etc)
  - ♦ Insert sound
  - ♦ Print
  - ♦ Links (with URLs, files, and other materials)
  - ♦ Saved with the corresponding model, saved in an appropriate format (for instance, HTML format)

The reports will be pre-prepared by teachers in an appropriate format, depending on the students and the task, in order to incorporate prompts that could support students.

It has to be noted that the 'Report' tool will also be extensively used during collaborative activities.

All the above requirements have to be considered as having:

Degree of necessity: ***Desirable***

Time of necessity: ***In second prototype***

Stability of necessity: **Flexible**

3.4 Study Themes Area

The 'Study Themes' is the area where the situations to model, or the problems to solve are presented.

- The space contains a text editor that allows students and teachers to prepare new problems or to modify existing ones.
- The editor should allow multimedia material: text, image, sound, and video.
- The synergy with other text editors (like Word) must be assured.

All the above requirements have to be considered as having:

Degree of necessity: ***Mandatory***

Time necessity: ***In first prototype***

Stability necessity: ***Stable***

3.5 Entities Editor

- The Entities editor, will allow the creation of new entities of different kinds.
- It will allow the definition of an entity, indicating the entity type, the entity name, the literal description of the entity, its properties, the names of these properties as well as their initial values.
- It will also support the assignment of concrete images to the various possible states of the variables. This is needed so as to produce the simulation of the state of the entity, when it is involved in an executable model (in the models-design and testing area).

All the above requirements have to be considered as having:

Degree of necessity: ***Desirable***

Time necessity: ***In the final application***

Stability necessity: ***Flexible***

3.6 Files manager

- The *"Files' manager":* implements a special view of the file system, hiding the file's real location on the hard disk and showing the hierarchy of the classes and their students/users instead. In addition, a special tree is 'assigned' to every user (student or teacher), which represents the various themes of study and the underlying created models. As a result, all a user needs to know to access his created models is their name in the system, the class they belong to, and the specific subject of study they are interested in. It must provide also a direct access to a disk drive or a CD recorder.

- File manager must provide a direct access to Repository (public and private ones).

All the above requirements have to be considered as having:

Degree of necessity: ***Mandatory***

Time necessity: ***In first prototype***

Stability necessity: ***Stable***

3.7 Help System

The "*Help system"*contains three sub-components:

- an on-line help (a hypertext system);
- an immediate context-sensitive help giving audio guidelines on using the various components, and
- an intelligent help system, which is activated during specific interface actions (during "decision making" modelling).

The above requirements have to be considered as having:

- The On-line help:

Degree of necessity: ***Mandatory***

Time necessity: ***In first prototype***

Stability necessity: ***Stable***

- The Context-sensitive help:

Degree of necessity: ***Desirable***

Time necessity: ***In second prototype***

Stability necessity: ***Stable***

- The Intelligent help:

Degree of necessity: Desirable

Time necessity: ***In second prototype***

Stability necessity: ***Stable***

3.8 Teacher assistance tools

3.8.1 Supervision (via local network)

The tool is a 'Viewer' that will support two main functionalities:

a. Supervise students screens when they work for instance in the local network
b. See in an easy way and simultaneously a number of "models' images": e.g. final models of students. These may also be the images of the models produced by the analysis of students' interaction history related to the applied modelling process.

   The second functionality helps teachers to have a quick view on students' productions, so as to compare these productions.

All the above requirements have to be considered as having:

Degree of necessity: ***Mandatory***

Time of necessity: ***In first prototype***

Stability of necessity: ***Stable***

3.8.2 Students' Interactions Presentation and Analysis Tools

The students' interaction analysis tools are an element of the system that allows the user (teacher and researcher) to obtain detailed information or even an elaborated and high level analysis about the way the model is constructed.

Three kinds of tools are proposed to be available to teachers:

1. Literal history of modelling Process

It presents the logfiles, which have been produced during the whole modelling activity, in a readable way. The inspection of the logfiles by teachers is a valuable activity specially during collaborative activities, when the interaction is based both on written dialogue and action.

The logfiles could be processed, in order to be readable by teachers, in the following ways:

a) by associating an icon code to each category of events:

For instance:
- ▣ Insert "Name entity"
- ▭ Insert "Name of Property"
- ⇔ Insert "Relation Name" , linked with "Property Name" and "property Name"
- >> Select "Variable name1" and "Variable name2, and "Variable name3" for representations
- ▸ Activate "Run'
- ↳ Activate Graphs
- ▪ Activate Table of Value
- ▤ Write into the notebook
- ● Write into the sticky note
- ✎ Write into the Report

etc.

b) Each item that is inserted but deleted later could appear under-highlighted, when inserted.

c) Some non-significant events (like intermediary "move item") to appear in smaller letters.

2. Visual History of Modelling process

This tool or functionality will permit teachers to select one instance of the logfile and make reconstruct the corresponding state of the models design area.

This functionality allows to re-construct instances or events that happened during the modelling process, and allows teachers, for instance, to understand the context of a dialogue extract.

All the above requirements have to be considered as having:

Degree of necessity: ***Mandatory***

Time necessity: ***In first prototype***

Stability necessity: ***Stable***

3. Modelling and Collaboration Activities' Analysis Tools

a) Object Oriented Modelling and Collaboration Analysis

The analysis of students' modelling process is based on the framework presented in Part III, unit 2.4. of the current document. From the resulting framework of analysis, a model M of the modelling process is defined and conceived as a formal analytical model, that analyses certain aspects of actions and dialogues (if modelling process was based on collaboration through network) occurring in the students group.

This model of the students' process is based on the notion of ownership of the components of the model provided by students. The entity/property/relationship constructs of the final model, as well as these inserted and then rejected could be one set of the basic objects that make a model. To each of these items a number of attributes can be associated, that concerns the order of its insertion, as well as a set of functional roles that are related to the eventual collaboration between at least two partners. The functional roles of low level students' actions reveal who has inserted the item, who has modified or contested it, who has argued on it, or who has tested one or more items using the available tools.

The analysis produces a model of students interactions that could take the following general form: Let a given Model S of a problem X be: S(X) = {E_i , R_j, A_m}, where E represents the node entities of the solution, (i=1, …, k), A the properties of the entities (m=1, …, n), and R the relationships connecting them (j=1,…,l).

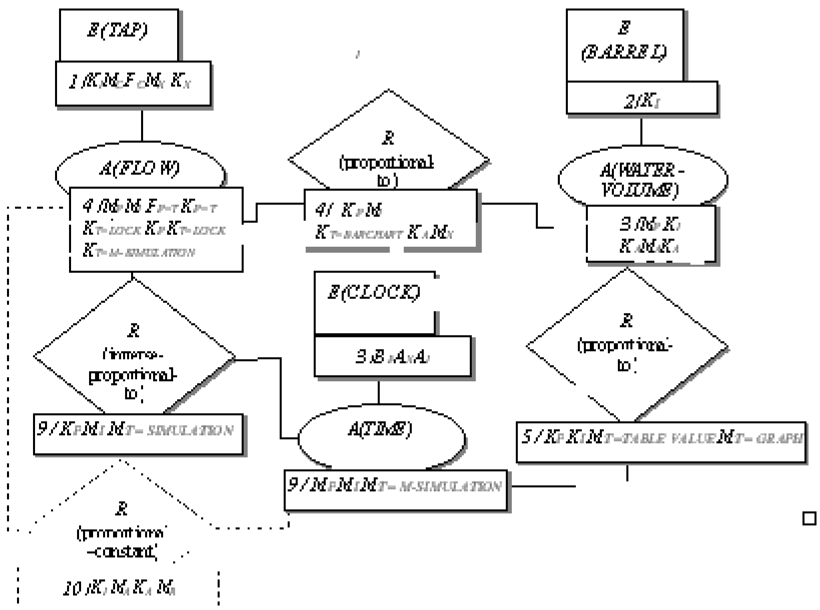
$$M\ (S) = \{ E_i\ * \tau i\ / P_i f_j,\ P_k f_l\ ,\ \ldots\ R_j\ * \tau i\ / P_i f_j,\ P_k f_l\ ,\ \ldots,\ A_m\ * \tau i\ / P_i f_j,\ P_k f_l\ ,\ \ldots\ ;$$

$$-E_i\ * \tau i\ / P_i f_j,\ P_k f_l\ ,\ \ldots\ ,\ -R_j\ * \tau i\ / P_i f_j,\ P_k f_l\ ,\ \ldots\ ,\ -A_m\ * \tau i\ / P_i f_j,\ P_k f_l\ ,\ \ldots\ \}$$

The analytical model of the model produced by students can be:

Where: E, R, A, are the entities, relations and variables that are part of the final students' model, and –E, -R, -A the items discussed during the modelling process, but not appearing in the final model. τ_i is an index of the item, as implied by its initial action of insertion or by its discussion in the timeline of the problem solving process.

The analytical model could take a diagrammatic form: such as:



b) Advanced analysis on students' basic functional roles:

Further analysis on the Model M of the students' modelling process can produce more sophisticated qualitative and quantitative indicators: such as the extend of use of 'Test' tools (run, graphs) by one students or by a students' group, the degree of participation, the density of interactions, etc.

The results of such as analysis could be presented by literal or diagrammatic mode.

All the above requirements have to be considered as having:

Degree of necessity: ***Mandatory***

Time necessity: ***In first prototype***

Stability necessity: ***Stable***

3.9 Students' meta-analysis tools

Some of the previous tools that present and analysis of the students' interactions could be designed in a way appropriate to be addressed to students.

1. Visual History of Modelling process/ Memory aid

This tool or functionality will permit students to see a playback of the modelling process, thus providing a memory aid.

2. Advanced analysis on students basic functional roles - Indicators on the quality of modelling and collaboration:

- Sophisticated qualitative and quantitative indicators: such as the extend of use of 'Test' tools (run, graphs) by one student or by a students' group, the degree of participation, the density of interactions, etc. could be derived and presented to students

These indicators could be presented by a literal or diagrammatic mode.

The exact form of interaction analysis tools that will be addressed to students will be determined after the initial investigations using the prototype with students and in different collaborative settings.

All the above requirements have to be considered as having:

Degree of necessity: ***Mandatory***

Time necessity: ***In second prototype***

Stability necessity: ***Stable***

3.10 Initial Specifications related to Learning Community and Collaborative learning sessions

3.10.1 General Community Level functions and tools

The website, its material, tools, and functions are considered in order to support in an appropriate way the whole "Learning Community".

We can distinguish: A) The website general level information and tools; B) The groups management tool; C) The session management tool (using also a 'presence info tool); D) The repository of material with a database allowing different kinds of materials.

It has to be noted that all actions and exchanges of the whole community must be recorded, given that this kind of data will be used for research purposes. For that reason, logfiles of community acts must be created and saved.

All the requirements presented here in after has to be considered as following:

Degree of necessity: ***Mandatory***

Time necessity: ***In first prototype*** *(in a first or simplified version)*

Stability necessity: ***Stable***

3.10.1.1 Website related information and tools:

- General level Information or Content: some basic educational material about modelling and models in sciences, sites presenting the schools and students groups involved, links to other interesting sites, etc.
- FAQS;
- discussion forums;
- a bulletin board for announcements;
- links with the repository of materials created from a specific group or community will be also provided.

The above tools and material will be available for each main community (this of students and this of teachers). Discussion forums and bulletin boards may also be available at a group level.

3.10.1.2 Communities and groups:

Two different main communities may be established: Students' Community, and Teachers' Community. Internal to each community different groups can be created.

For all the users' categories a login password is needed in order to enter into the corresponding Community Space, and in order to login and have access to the specific Group Space.

All the registered schools, groups' names or individual users (not belonging yet in a specific group) will be presented at the community level. At the group level, more detailed information concerning individuals participating in groups will be presented. [Profile (Student, teacher, administrator), full name, user name, IP address, school name, class, telephone number, photo, interests, etc.]

(A 2D spatial metaphor can be used, if necessary, for presenting students and teachers from each school.)

It is to be noted that a videoconference application will be available and it could be used for specific events and sessions to help "to get to know" another group - specially for sessions that will initiate a more closed cooperation/collaboration between schools.

3.10.1.3 Group management and collaborative sessions

Group management: Facilities for establishment and management of groups should be provided. The 'group management' will provide an 'address book' tool, and group definition/creation facilities.

Groups should be made of registered users.

- Two or more users can be defined as belonging to a group, using the users' local *address book* or the *users' list* (these users must already belong to the one of the two communities). Thus, a group is defined. In the same group, users with different profile can participate.
- A group can be modified (e.g., by adding a new member or 'removing' an old one)

It should be noted that Groups can also form in the classroom (eventually using the school server). There are no restrictions in the profile, or identity of users belonging in a group.

Users that have access to the 'group management' are mainly administrators and teachers. However, the possibility to create a group (in order to proceed to a collaborative session) must be also offered to the students.

To summarise, the group management tool, will support:

- communities management;
- groups management;
- individual users' management.

At the level of users, it will support:

♦ user registration;

♦ user login;

♦ identification of the state of the user (on-line or off-line).

Group session: A group session tool that permits the user to register in a collaborative learning or teaching scheme and indicates the beginning of the activity. This tool should support functions as the following ones:

♦ initiate a session;

♦ finish a session;

♦ quit a session (by a member);

♦ join a session;

♦ etc.

Group session must support both sessions for synchronous and asynchronous collaboration. In both cases, there are no restrictions related to the number of session members.

It should be noted that:

⇨ Students must also have the possibility to activate a 'group session'.

⇨ Teachers have more rights (e.g. follow a session as an observer, follow more than one session in real time, etc.)

(In the project's technical annex, The "Group Session mechanism" is considered as "Meeting Planner")

The group session too, will have a synergy with a 'presence tool' presenting all the 'connected' persons at each time.

***Presence info tool*/function**: It presents all the connected persons or groups, by presenting their names, and if asked also, other identity elements regarding for example their group, school, class, etc.. Queries could also be supported (e.g. show all students, or show all teachers).

3.10.1.4 Repository of material

The material stored in the repository can be: primary entities, models (problem solutions), problems (themes of study), reports on modelling process, as well as logfiles. However the repository must be expandable, allowing one to add new categories of material (e.g. general purpose materials such as: txt documents, video documents, images, etc.).

Every 'material' apart from its name and type has as identity the date of creation, as well as the name of its author. From this last element, name of author, supplementary information can be derived (IP address, name of school, profile (teacher/student), etc). A 'material' could be also characterized by the author, as 'public' or 'confidential'. A 'public' material could be stored (uploaded) at the community level, while a 'confidential' material, remains in the individual or internal to group space.

We could imagine this repository as a 'database', where a number of queries must be supported, such as the following ones:

- New problems (models, reports, etc.) posted by a specific group, or by a specific user.
- List of all new materials posted per category, during previous two weeks.
- A table presenting all materials at the community level, and/or at the group level, and/or at the user level.
- All materials of one (or more) category for all the communities, groups and users

The groups should also have their own repositories of material: problems, models, reports, histories of interaction, etc.. The students could have access to this repository, so as to save material, open or download material, etc. Teachers should have supplementary rights, such as file name modification, or deletion of a specific material.

It has to be highlighted that in all the cases the flexibility of use in school or at home must be supported and specially for materials that are under development.

3.10.1.5 Logfiles of Community events and acts

Logfiles will be automatically created and saved, presenting every action registered.

3.10.1.6 Awareness of members' previous actions

Inside the area of each group or community, a table could provide by default information about the members' last actions, exploiting info either from database of material and/or from website logfiles.

It can present information such as: a 'file is saved', by 'name', date, author. The information will be presented to each member of the group entering to this area. This functionality will permit a member of a collaborative group that work asynchronously, to be informed that his/her partner has worked and for instance has saved another version of the 'X' model, the previous morning. This kind of information, if automatically available, gives the possibility to all members of group to acquire an "Awareness of members' previous actions", each time they login and enter in the group space.

It is very important for asynchronous collaboration sessions, but also for members that work in synchronous collaborative modes.

This functionality corresponds to the "Others Activity Info" described in the project's technical annex.

3.10.2 Collaborative Learning sessions related tools

All the above requirements have to be considered as having:

Degree of necessity: ***Mandatory***

Time necessity: ***In first prototype*** *(eventually in a simplified version)*

Stability of necessity: ***Stable***

3.10.2.1 A shared workspace

The shared workspace must support synchronous collaborative work on the models design area, as well as the use of all the annotation tools (sticky notes, notebook, and report).

The private space exists in parallel, and partners can open another model in another window.

It should be noted that the above distinction between private and shared workspace must be taken into account in the unified logfile of a collaborative session.

The transition from the private workspace to the shared workspace, must be supported by direct manipulation (select one or more items, drug and drop).

3.10.2.2 Group members' info tool:

It provides brief information about the users participating in the session. It is a part of the 'Presence info' tool. It presents mainly the name (or nick name) of a user and if they are actually on-line or off-line.

3.10.2.3 Action Co-ordination tool of the shared workspace

It permits to manage the actions that take place in the shared workspace by the partners during model's creation. It concerns synchronous collaboration.

It allows partners to coordinate their actions, through an *Action* Key protocol of coordination. One takes the 'control'; then others need to ask for it via dialogue tools, in order to get it.

It is to be noted that this 'control' concerns only the functionalities of the main space of model design and not the whole environment. So the other partner can see the representations, or open a theme of study, etc.

One of the matters that is related to the protocol of collaboration has to do with the rights of modification or deletion of an item of solution inserted by one of the collaborators. It is argued that between the collaborators, only the person that has inserted the object has the right to modify or delete it. The person that has inserted one or more items can give the right to his partner to delete or modify one or all of them. The teacher has also same right; s/he could intervene in cases of impasse.

In order to facilitate a good collaboration without discrepancies, the system could *present* in an immediate manner *the ownership* of each solution item (the name of the person that has inserted it, or that has deleted it).

In order to facilitate the W*orkspace Awareness of other members' actions* when they collaborate in a synchronous mode, specific functions may be available: representation tools activation, theme of study activation, entities library activation, etc.

3.10.2.4 Collaboration-dialogue based tools

Three different kinds of dialogue tools must be available: an appropriate chat tool, linked and unlinked sticky notes, as well as structured chat dialogue. In some sessions some tools may be deactivated depending the session options defined by the teacher. Different tools will be useful/appropriate during different problem solving and modelling situations or different collaboration settings.

a. **Chat tool:** A chat window can be shared respecting the WYSIWIS (What you see is what I see) protocol. It can have three areas: one for the student's current contribution, one for the incoming one and one for the history of chat dialogue.

b. **Structured dialogue tool** with sentence opener interface: The dialogue interface must contain sets of sentence openers (e.g. "I think', "I agree because"), organised in intuitive categories. To contribute to the dialogue, the user must first select a sentence opener. The selected sentence opener may appear in a text box below the group dialogue window (presenting the history of the dialogue), where the student may type in the rest of the sentence. One or more of the sentence openers, will refer to questions addressed directly to the teacher, asking for help. During synchronous collaboration, the teacher who supervises simultaneously a number of groups, must be informed that s/he has received a message (by sound or by a blinking message).

c. **Sticky notes:** Sticky notes allow to add comments directly into the model design area. They may concern a specific item of solution (entity, property, or relation) or a part of the whole solution. A sticky note can be free posted in the area or linked to the specific object, or to an area. One or more items must be selected before or after writing a note. Another partner can also post another sticky note referred to the same object.

d. **Unified and linked dialogue story**: All the three tools can be linked and embedded, in order to preserve the chronological history of the dialogue. Thus, a fourth tool may be needed in order to record and re-present the whole story of dialogue, in a chronological order.

In order to provide a logical linkage between the two artefacts (sticky note and free or structured dialogue), the dialogue based on sticky notes must appear in the unified dialogue story together with the name of the item or of the group (e.g. "<Kostas> (related to 'item name') Why…..").

4. Concluding remark

The design of a new technology-based learning environment is specially complex. The interdependency of pedagogical and technical development work appears to call for a close integration of these two aspects in the development of ModellingSpace. Software and interface design and development cannot succeed without a very close interaction with the pedagogical aspects of the project during the entire development lifecycle.

A serious risk the ModellingSpace project shares with all others in which innovative technology is used to create learning environments, is that both software and users-interface designers, and the pedagogical research community have to think beyond their own priorities, in order to avoid forming distinct and closed communities that only ask for formal input from each other. Instead they need to look outward, so as to create a community of mutual understanding based on reciprocity and commitment. Given that pedagogical design principles are always extremely complex and their applications may vary from one context to another, the only way of managing the problem is to create a really unified community of technical and pedagogical experts.

Therefore, the most important challenge is to establish an intensity of interactions and commitment from the two communities, which goes beyond the mere production of the deliverables in order to further detail the specifications, in a well-argued and grounded way, so as to ensure that design solutions actually improve quality of learning.

APPENDICES

APPENDIX A

Collaborative Problem Solving using an Open Modeling Environment

C. Fidas[1], V. Komis[1], N.M. Avouris[1], A Dimitracopoulou[2]
[1]University of Patras, Patras, Greece
[2] University of the Aegean, Rhodes, Greece
fidas@clab.ee.upatras.gr, komis@upatras.gr, N.Avouris@ee.upatras.gr, adimitr@Rhodes.Aegean.gr

APPENDIX B

OCAF: An object-oriented model of analysis of collaborative problem solving

N.M. Avouris[1], A. Dimitracopoulou[2], V. Komis[1], C. Fidas[1]

[1]University of Patras, Patras, Greece
[2]University of the Aegean, Rhodes, Greece
N.Avouris@ee.upatras.gr, adimitr@rhodes.aegean.gr, komis@upatras.gr, fidas@ee.upatras.gr

APPENDIX C

Towards open object-oriented models of collaborative problem solving interaction

A. Dimitracopoulou[1], N.M. Avouris[2], V. Komis[2], C. Fidas[2]

[1]University of the Aegean, Rhodes, Greece
[2]University of Patras, Patras, Greece
adimitr@rhodes.aegean.gr, N.Avouris@ee.upatras.gr, komis@upatras.gr, fidas@ee.upatras.gr

APPENDIX D

Preliminary findings from a pilot study on collaborative learning, in real school context, during simple problem solving.

Brief research report, prepared by

Argyro Petrou

Computer Science Teacher, PhD student, LTEE lab, Aegean University

1. Aims of the study

Most of the CSCL studies have concentrated on students' learning processes; the tutor's role has been much less often investigated. Therefore we want to:

- gather data concerning how teachers, given their pedagogical expertise and students' varying skills, attitudes and learning experience, are able to function within network collaborative learning environments;
- find out how can teachers be helped to help students in network collaborative learning situations;
- address the teacher's role in networked collaborative learning situations.

2. Settings

2.1 System

The approach uses:

1. Netmeeting which:
   a. Allows members of one team to exchange messages.
   b. Allows members of one team to use any program in common. Each time only one member of the team has the control. At the top of the shared program is written the name of the user that controls the program at that moment. The first one that asks for the control must click *Request Control* of the menu *Control* at the window of the shared program. If another member of the team wants the control he must repeat the process, but previously the member that had the control must click *Release Control* of the menu *Control* at the window of the shared program.
   c. Save the contents from the chat session to a file for future reference.

   When we use NetMeeting and we start a program that will be used in common from a team, the member of the team that is sitting at that computer has more rights. So we use two computers, one for each team, and we start the application from there. Noone sat at those computers so all the students had equal rights.

2. **NetSupport** which was installed at the teacher's computer (Control) and has students' computers as clients. With this program:
   a. The teacher (Control) can *share* a student's (client's) screen. The student's screen will be displayed at both the Control and the Client. Both the teacher and the students will be able to enter keystrokes and mouse movements.

b. The teacher (Control) can *Watch* a student's (client's) screen. Only the user at the client (student) will be able to enter keystrokes and mouse movements.

c. The teacher (Control) can watch multiple students' (clients') screens, each in its own window.

3. **CORIOscan Select, a computer to video converter**. Because we couldn't have the actions of the students at the shared workspace, and the actions of the teacher at his computer during the lesson, we used this device. It gave us the ability to transfer the image from one desktop computer to a standard video recorder. We connected one to each of the computers from which we started the shared application, one for each team. In this way we kept the actions at the shared workspace and the dialogue history in parallel for each team. We also connected one to the teacher's computer so we kept log of the teacher's actions during the lesson (e.g. how he used NetSupport).

2.2 Classroom Environment

The participants were ten children of sixteen years old, from two different classes. Five from each class. They were divided into two groups of three children and two groups of two children. The pupils were placed into mixed ability groups by the class teacher.

NetSupport and NetMeeting ran on eight locally networked PCs under Windows '98 for each class. Five computers for each student, one for the teacher and two computers, one for each team, where the program that was used in common started and no one sat at those computers (as we explained above).

Before started, the children received a short lesson (20 minutes) on how to use NetMeeting.

2.3 Task

Students worked on three activities (simple problem solving) from the lesson *Computers' Programming*, for six instructive hours (6 * 45 minutes) each class. The activities were not designed for the purpose of the study. They were chosen from the teacher, the students should do these activities anyway.

For the first activity, we asked from the students to comment on a given Pascal program and to justify their opinions. We asked a text written in common from each team. The shared workspace was a Microsoft Word document.

For the next two activities, one common program (written in Pascal) was required from each team. So, the shared workspace was the environment of Turbo Pascal. In the third activity we additionally asked from the students to answer a question, that is again a text written in common.

3. Preliminary Results

3.1 Teacher's role

Data from video recording (history of dialogues between students and between students and the teacher, history of actions at the shared workspace for each team and history of teacher's actions), camera recording and semi-structured interview with each teacher.

We have studied two possible scenarios for teacher intervention:

3.1.1 On-line

The teacher watches the students' interaction (dialogues and actions at the common workspace) on-line and acts as:

Provider of information

- The teacher intervened when the students couldn't find a mistake or when they didn't know how to continue. According to teachers' opinion, "*these are activities that must be completed, they are not a test on students' knowledge. You must activate them, to learn by doing. So you leave them a period of time and then you intervene. Also we must have in mind that the students must learn certain things during the school year and we don't have unlimited time*".
- The teachers intervened when the students were asking them to do so. In most of the cases they intervened without noticing if the subject had been discussed previously between the members of the team. They did that because *"that is how we are used to work until now since we didn't have the possibility to be familiar with the processes that the students use in order to solve the problem".* In some other cases the teacher told the students to ask first the other members of the team and in one case the teacher sent them to read the corresponding paragraph of their book.

Regulator of students' interaction

- **In the team**: the teachers urged the students to collaborate and to participate when it was necessary (especially in cases where from the study of previous dialogues and actions at the common workspace, a weakness had been noticed).
- **Between teams**:
    - **Exchange of members**. When one team had finished the activity and the other team couldn't continue, the teacher instead of showing them how to proceed, he took the student that found the solution from the first team and put him in the second team. At the same time, he took a student from the second team and put him in the first team. In this way he didn't intervene directly, he let the students exchange opinions, ***student-student tutoring****.*
    - **One team criticizes the solution of the other team**. Especially in cases where "logical" mistakes existed in the program, or in cases where no objective criteria existed about the solution of the problem, ***learning from examples.***

3.1.2 Off-line

At the end of the lesson, the teacher had the dialogues between the members of one team and the actions at the common workspace. Until then, the teachers were not familiar with the processes that the students used when they worked together. Now, they had the possibility to study the messages that were exchanged between the members of each team, the contribution of each student at the final common product, students' views. So the teachers should be made aware of these new possibilities and design the appropriate activities that make worthy the system, *"have in mind the new possibilities and design the appropriate activities".*

This new possibility changed the way of intervention and teaching:

- The teachers had the possibility to comment on students' participation and quality of collaboration.
- They didn't simply try to fix the mistakes at the final product, as they did up to that time. They did interventions that were based on the study and the analysis of students' dialogues and actions at the common workspace, *"you can see each member individually in the team", "the teachers have the possibility to inspect the way a specific team works and collaborate", "to find out the misconceptions".*

3.1.3 Comments

- At the first lesson, one of the two teachers seemed a little 'lost". He didn't watch the dialogues and intervened only when a student asked for it. From the second lesson he didn't have any problem.

- The minimum help was asked from the students as far as the collaboration is concerned (only once), even when they had some problems they didn't ask for help. They tried to solve them by themselves.
- They asked for help about the object of study only when no one from the team could give an answer. In most cases, when the teacher was present, the students tended to ask him and not their classmates. This happened because the particular children had learned to collaborate (two or three children work in front of one computer in most of the lessons).
- The fact that the teacher commented on the quality of the collaboration of a team, not in private, encouraged those that had good collaboration to continue like this, and the rest to become better.

3.2 Design of systems that support collaborative learning.

Data from video recording (history of dialogues between students and between students and the teacher, history of actions at the shared workspace for each team and history of teacher's actions), camera recording, semi-structured interview with each teacher and semi-structured panel interview with students.

In general, the teachers and the students liked the fact that they needed the minimum training in order to learn to use the systems. Also, teachers liked the fact that the program was content-independent, that they could use the program that they were using before in the classroom for the specific lesson, that they needed to change almost nothing.

3.2.1 Students' tools.

The system must provide the following possibilities:

- Display the names of students that are working together at that time.
- Exchange of messages between students of the same team. Also the system must notify the student that he has a new message, *"some times we are concentrated on the program and we don't see the new message".*
- Exchange of messages between teams when the teacher allows it, *"we think that the communication between teams is very important"*.
- Exchange of messages between students and the teacher. Also the system must notify the teacher when a student sends a message to him, *"we send a message to the teacher but he didn't see it".*
- The students must be able to press buttons for some standard messages, "*it is very time consuming to write messages, if could avoid typing some standard ones, like I don't understand, Yes, No, I want the control…*".
- Coordination of messages. A lot of times a member poses a new question and another member answers the previous ones. This happens especially in teams with three members. For example, if the system can display under the name of each student what he is doing now, typing, having the control of the common workspace, etc., as part of the **Awareness of students' activity tool**.
- The students must be able to work at a shared program. Of course the system must coordinate the access at the shared program in a way (***Coordination tools for the shared workspace management***)*.* The way NetMeeting coordinates the access to the common workspace causes us no problems. The only think that we can add is that when a student asks for the control and some time passes and he hasn't got it, the system must remind the team that the specific student has asked for the control. Or the system can provide the information of who has asked for the control, as *part of* ***Awareness of students' activity tool****.*

- The students must be able to work their shared program in full screen; the chat history should be displayed at the edge of the screen, *"we couldn't have our program full screen, because it covered the messages".*

3.2.2 Teachers' tools.

The system must provide the following possibilities:

- The teacher must be able to watch the chat history and the actions at the shared workspace for each team.
- The teacher must be able to watch the chat history and the shared workspace for as many teams as s/he wants at the same time.
- The teacher must be able to intervene, send a message or take the control of the shared workspace, when s/he thinks it is necessary.
- The teacher *"must be able to show the shared workspace of one team to another team".*
- The teacher *"must be able to allow two or more teams to communicate when he thinks it is necessary".*
- *"A log file must exist with the dialogues (between students and between students and the teacher) and the actions that take place at the shared workspace in parallel for each team"*. So afterwards, the teacher at any time can have the messages that have been exchanged up to that time and the picture of the common workspace at that time, (***Dialogues and shared workspace actions recording tool***). Also,
  - ⇨ "*The contribution of each student to the final product must be obvious*", for example different colour for each student in a team. "If something has been deleted at the common product, it must be crossed over" with the colour of the student that deleted it, (part of **students' work analysis tool**).
- The system must model the dialogues in a way, so the teacher is helped while s/he is watching the dialogues on-line but also afterwards at the analysis of the dialogues: *"May be the system must build a forest with trees from each dialogue, in this way, for example, you can see easily which question hasn't been answered" (**Visualization of dialogues**).*

3.3 Teachers' views.

Data from semi-structured interview with each teacher. It has taken place at the end of all sessions.

According to teachers' views, this new way of collaboration:

- Imposes more discipline.
- It is a more cold medium, *"when you press enter everything finishes*".
- When it is compared with the situation where two or three students are sitting in front of the same computer, this way reduces social comments: "*there isn't much time, so the social comments are reduced"*.
- Gives the possibility to a teacher "*to inspect the way a specific team works and collaborate, to locate where there are misconceptions, in order to intervene*". The teacher *"can reproduce in the classroom snapshots from the problem solving process, not just the final product when he comments with the students on the way they solved the problem*".
- *"Forces*" students to write which is positive and something that they aren't doing very often in other lessons. We noticed that students had difficulties when they were trying to formulate their questions, they were not familiar with the words, the terms that they should use, because they should be accurate and brief - *"the students became more familiar with the terms".*

- Allows a teacher to watch four or five teams at the same time. The number of the teams is depended on the task; here the whole process is slow (typing of messages and programming are time consuming processes). Of course, if the task changes and *"the student can solve the problem with 'drag and drop' for example, then it will be impossible for the teacher to watch more than two teams at the same time".*
- Is suitable in cases where *"the teacher wants to see what the students haven't understood, what the misconceptions are, what is the contribution of each student in a team, how a team works, but of course not for all the instructive hours of a lesson".*
- *"Would be interesting to be applied between students that are at different classes or even at different schools*. *Also in the afternoon, after school, where students are at their homes".* A lot of times, students want to be met in the afternoon in order to solve the problems together, but most of the times they can't because of the distance and their obligations.

3.4 Students' views

Data from semi-structured panel interview. It has taken place at the end of all sessions.

According to students' views:

- This new way of collaboration reduces the social comments, "*it is a more task focused interaction*" - "*it is not easy to write about your motor bicycle for example, you write about the problem*".
- In this way, students "*communicate and understand each other better*".
- Because they can have the control of the shared program when they want, they "*feel more responsible*", they "*want to participate*".
- They don't like very much "*the typing of a message, is so time consuming*".
- They don't like the coldness of the medium.
- Some of them feel embarrassed to express their questions because the teacher is watching the messages, "may be I ask something stupid".
- Some of them believe that because the teacher is watching the messages, "more students will participate".
- They "want to collaborate with other students" from their homes "in the afternoon".

Acknowledgements: We acknowledge students and two teachers of the 3rd Lyceum of Rhodes (Greece), who have participated in the experimentation in December 2001. Also, we acknowledge George Fessakis, for his encouragement and assistance.

APPENDIX E

Design of collaboration-support tools for group problem solving

C. Fidas[1], V. Komis[2], N.M. Avouris[1]
University of Patras
[1]Electrical and Computer Engineering Department - HCI Group
[2]Early Childhood Education Department
GR 26500 Rio Patras, Greece
Fidas@ee.upatras.gr, Komis@upatras.gr, N.Avouris@ee.upatras.gr