A LEARNING ENVIRONMENT FOR DIGITAL ELECTRONICS

Giuliano Donzellini[1] and Domenico Ponta[2]

[1,2]DIBE - Department of Biophysical and Electronic Engineering
University of Genoa, [1]gdonzie@dibe.unige.it, [2]ponta@dibe.unige.it

ABSTRACT

Deeds is a learning environment for digital electronics, providing tools and resources to supports teachers and students. The Deeds three simulators cover combinational and sequential logic networks, finite state machine design, microcomputer interfacing and programming at assembly level. They are integrated together, allowing the design of state-of-the art digital systems, composed of standard logic, state machines and microcomputers. Deeds guides student's activities by delivering learning materials through a specialized browser. The environment has been specifically designed to support distance education and project work within an inter-institutional and international context. A collection of learning materials based on Deeds, covering an introductory course of digital design, is available. Deeds is currently used in our courses, as the main tool to implement the "learning-by-doing-approach".

1. INTRODUCTION

The DEEDS (Digital Electronics Education and Design Suite) is a simulation-based learning environment for digital electronic design. Our team at DIBE – University of Genoa, Italy, has developed it as part of its research activities in the fields of distance and cooperative learning.

We developed Deeds specifically for educational applications, with special attention to the needs of the courses of the first years of the Information Engineering courses. Its rationale is the application of the constructivist methodology, where the necessary knowledge and skills develop in the learner as a result of a progressive exploration of the technical issues of the courses. To achieve this goal, a set of simulators and related resources provide the students with the logical tools that allow the implementation of the above mentioned pedagogical methodology [1].

Deeds helps students in acquiring the theoretical foundations of digital design, together with analysis and problem-solving capabilities and practical synthesis and design skills. It is now extensively used by the students of the first year of information engineering and as a support for project-bases courses. Deeds covers the following areas of digital electronics: combinational logic networks, sequential logic networks, finite state machine (FSM) design, microcomputer interfacing and programming at assembly level.

Deeds lends itself very well to different formats of instruction (lectures, exercises, lab assignments, etc.) and adjusts to different student levels. This is possible because Deeds is based on a set of tools that teachers can combine together and personalize to suit their pedagogical needs by contributing to the lecture space their own learning materials. The simulation tools themselves may adapt to different student level and provide a subset of their features when used with beginners.

The use of an Internet-controlled electronics lab, developed in our department [2], adds the possibility of performing real hardware experiments.

2. THE DEEDS LEARNING ENVIRONMENT

The Deeds learning environment includes three simulation tools: a digital circuit editor/simulator, a finite state machine editor/simulator and a microcomputer board emulator. All the simulation tools are characterized by a “learning-by-doing” approach. They are fully integrated together: design and simulation of complex networks integrating standard logic with state machines and microcomputers are therefore possible.

The simulators of the learning environment are accessible from HTML browsers that enables active Internet navigation to sites where students can find pages with lessons, exercises and laboratory assignments.

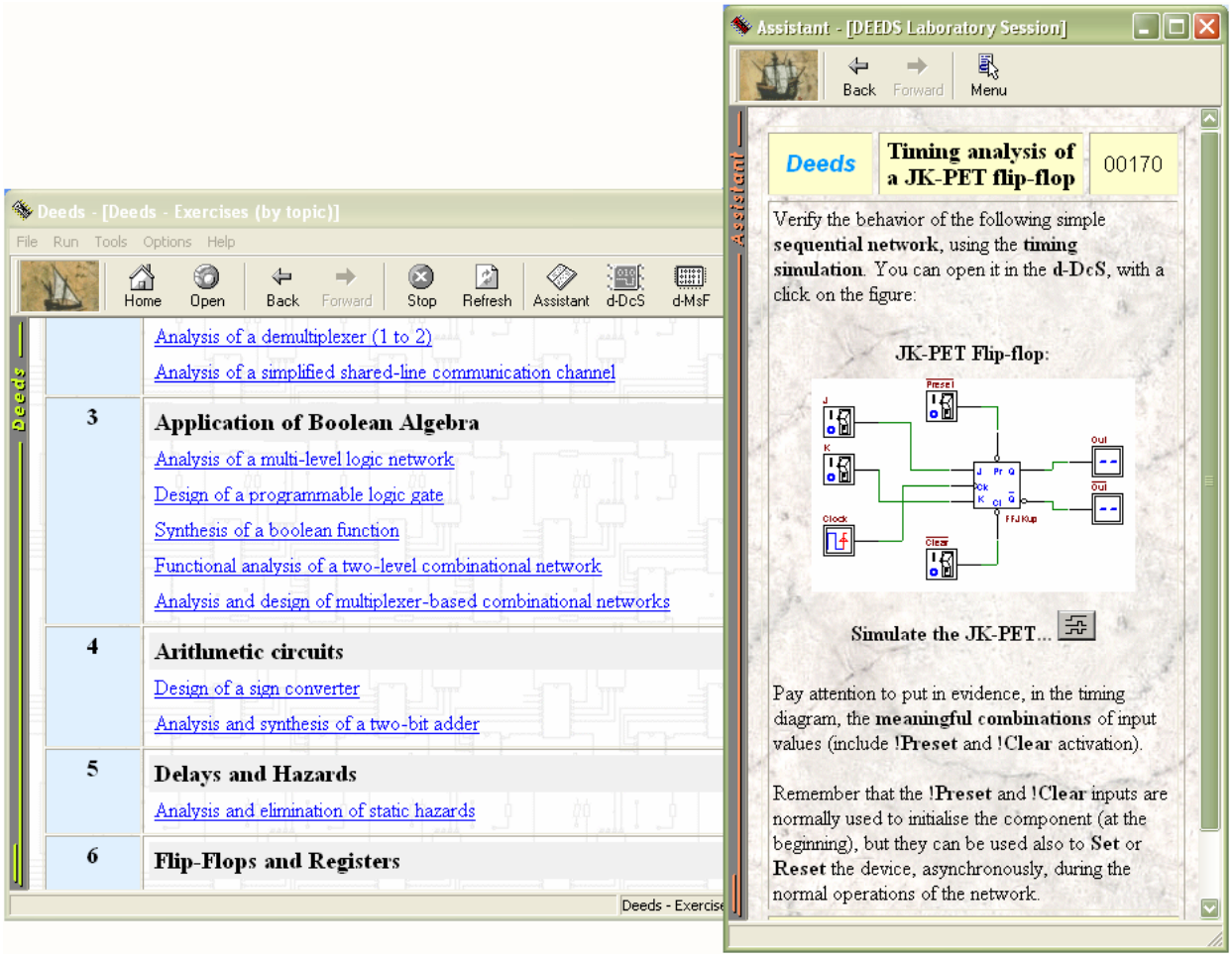

Fig.1: On the left hand side, the Deeds main browser, which connects to the web sites containing the learning materials and, on the right, the Assistant browser, which assists students side by side in their work.

The Deeds main browser is used to connect to the sites containing the learning materials, and the other one, the ‘Assistant’, is designed to assist students in their work (Fig. 1). The browsers, based on the standard MS Internet Explorer component, support all the features that the user can expect to find, including JAVA, JavaScript, VBScript, XML support.

Teachers do not need specific authoring tool, because the lecture space can be composed with any HTML editor. An editing-helper application that facilitates the linking of the editors and simulators commands to the text of the lecture is under development.

The graphical schematic editor for digital networks is based on a library of basic logic devices, that includes user-definable components and a simplified microcomputer board. The user-defined components are designed as FSM and can be built with a dedicated graphical editor/simulator. Once completed, the user-defined FSM can be inserted in the digital schematic and connected to standard logic digital networks.

Also a microcomputer board can be part of the digital system under development. The board makes available standard input-output parallel ports, besides other inputs as clock, reset and one interrupt request. The board must be programmed at assembly language level. Using standard logic ad/or FSM, the schematic editor allows building specialized input/output devices that can be attached to the microcomputer board.

Simulation of simple digital networks (gates, flip-flops, pre-defined combinational and sequential circuits) is allowed, as well as simulation of FSM, with runtime display of the relation between state and timing evolution and the contemporary execution of programs running on the micro-computer based unit. In this way, the user can simulate complex systems composed of standard logic, FSM blocks and an embedded microcomputer unit.

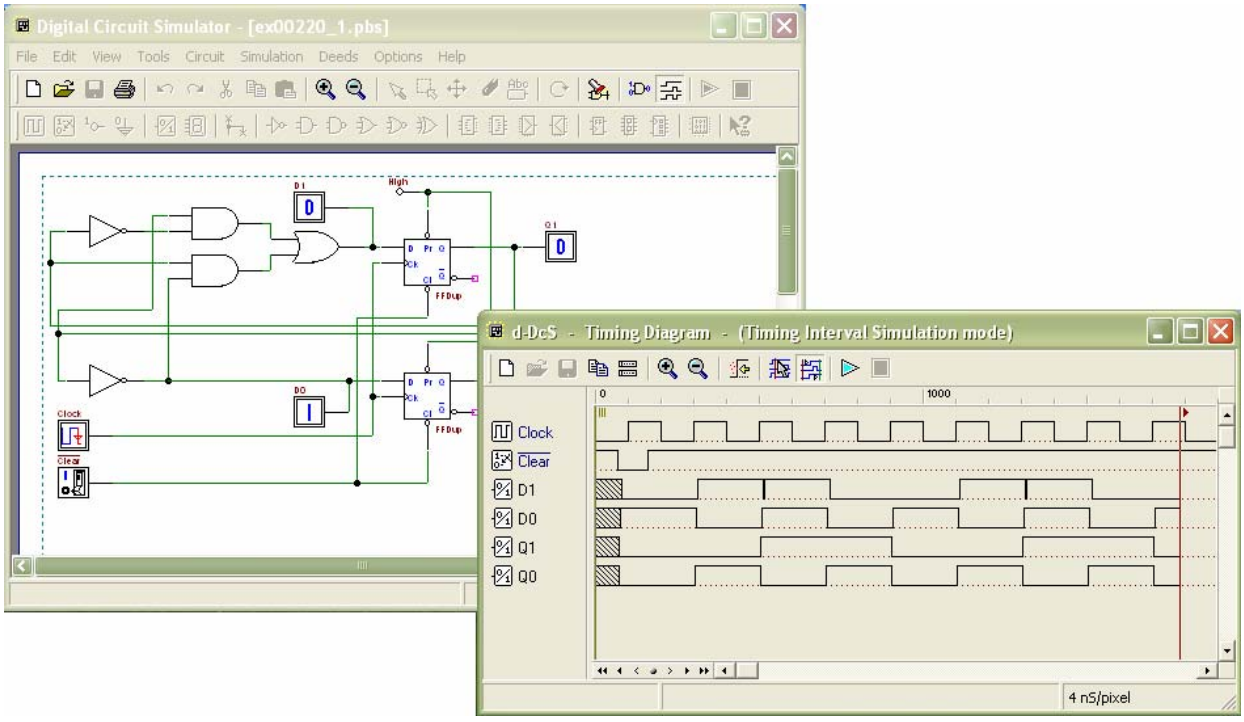

Fig.2: A digital circuit opened in the Digital Circuit Simulator and, on the foreground the timing diagram resulting from the simulation of it.

2.1. The Digital Circuit Simulator (d-DcS)

The Digital Circuit Simulator appears to the student as a graphical schematic editor (Fig. 2), with a library of simplified logic components, specialised toward pedagogical needs and not describing specific commercial products. As described before, the schematic editor allows to build simple digital networks composed of gates, flip-flops, pre-defined combinational and sequential circuits, custom-defined components (defined as Finite state machine) and a micro-computer board.

Simulation can be interactive or in timing-mode. In the first mode, the student can "animate" the digital system in the editor, controlling its inputs and observing the results. This

is the simplest mode to examine a digital network, and this way of operation can be useful for the beginners. In the timing mode, the behaviour of the circuit can be analysed by a timing diagram window, in which the user can define graphically an input signal sequence and observe the simulation results. This is the mode nearest to the professional simulators.

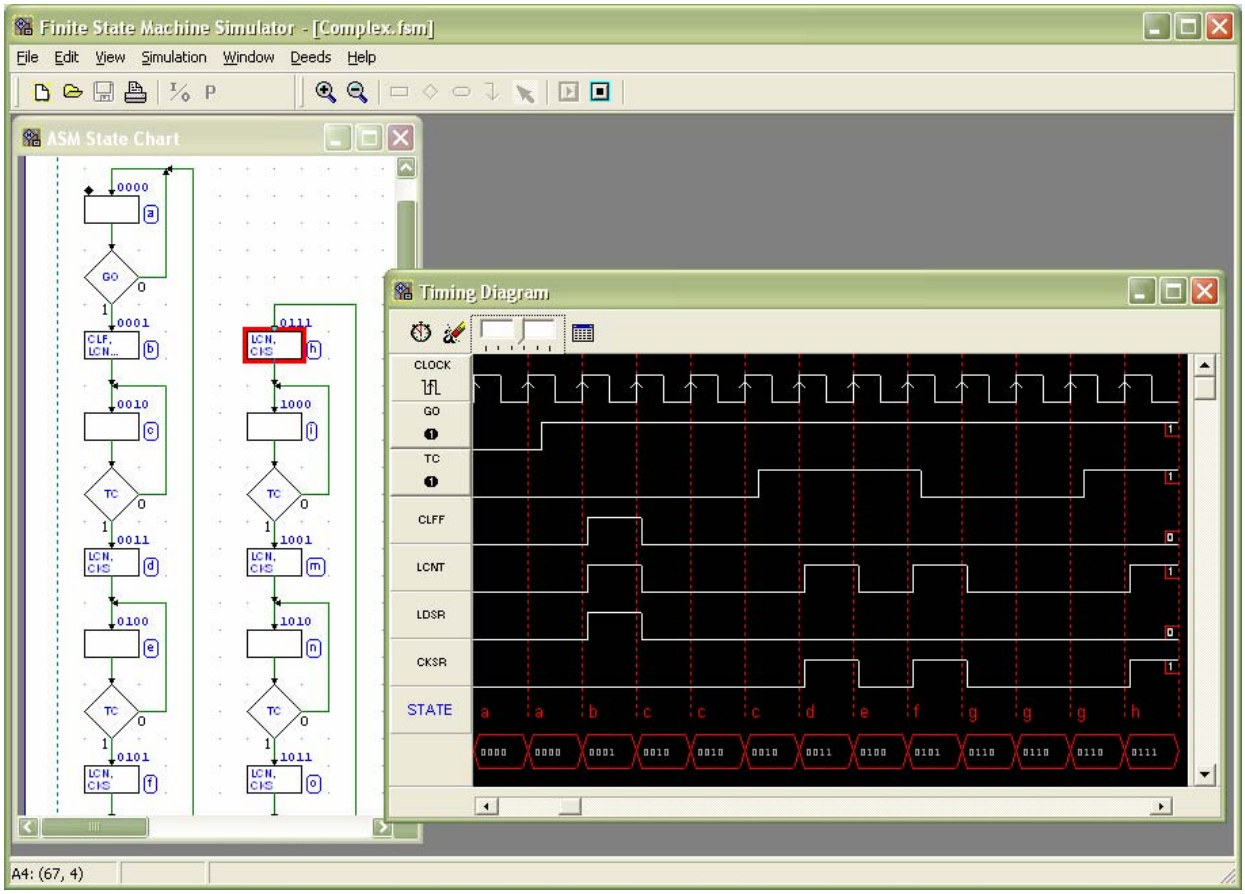

Fig.3: The Finite State Machine Simulator at work. The editor, on the left, shows the Algorithmic State Machine description. The timing diagram, on the right, is the functional simulation.

2.2. The Finite State Machine Simulator (d-FsM)

The Finite state machine Simulator allows graphical editing and simulation of Finite state machines components, using the ASM (Algorithmic State Machine) paradigm or a hardware description language approach. The tool allows the local functional simulation of the finite state machines designed by the user, with runtime display of the relations between state and timing evolution (Fig. 3). The components that the d-FsM produces can be directly used in the d-DcS and inserted into any digital circuit. Also, it can be exported as VHDL process.

2.3. The Microcomputer Board Emulator (d-McE)

With the Microcomputer Board Emulator the user can practice programming at assembly language level (Fig. 4). The functionally emulated board include a CPU, ROM and RAM memory, parallel I/O ports, reset circuitry and a simple interrupt logic. The custom 8 bit CPU, named DMC8, has been designed to suite our educational needs, and it is based on a simplified version of the well-known 'Z80' processor.

The integrated source code editor enables user to enter assembly programs, and a simple command permits to assemble, link and load them in the emulated system memory.

The execution of the programs can be run step by step in the interactive debugger, where the user can observe all the structures involved in the hardware/software system.

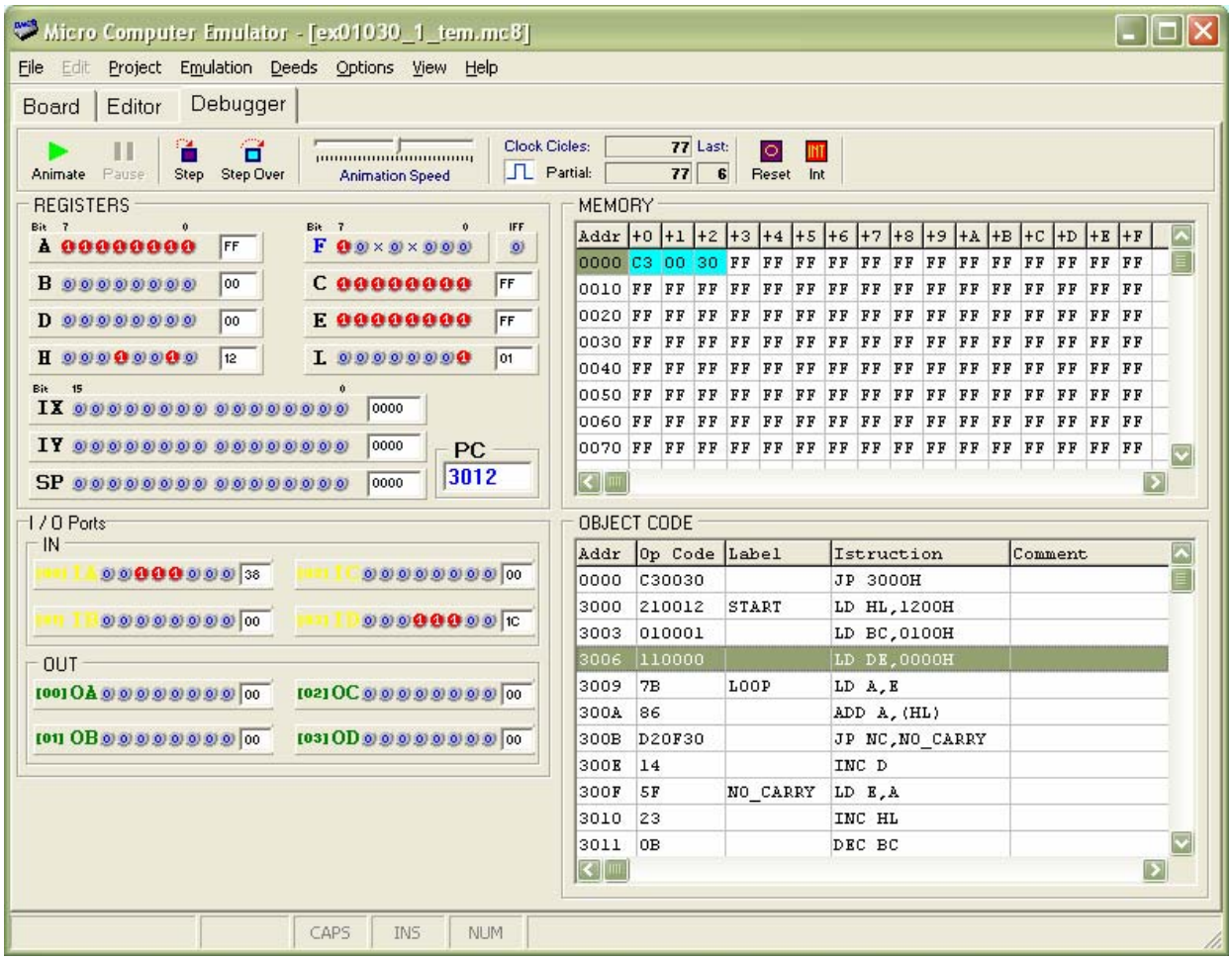

Fig.4: This screen shows the “Interactive Debugger” of the Microcomputer Board Emulator (d-McE) while executing a program step by step. The contents of all the microcomputer structure are displayed.

3. USING THE DEEDS

3.1. The Deeds approach to theory

A lecture based on Deeds appears as an HTML document with text and figures. Text, figures and visual objects can be active, because they are ‘connected’ by the browser to the editing and simulation tools of Deeds. For example, let’s suppose that the theory presents a digital circuit, and displays its schematic. When the user clicks on the schematic, Deeds launches the corresponding simulator, and opens that schematic in it. As necessary, the Deeds open another browser (the Assistant) that can contain step-by-step instructions on how to design, explore or test the circuit itself.

Such procedure is equally useful to convey concepts both on simple components and on more complex networks.

3.2. The Deeds approach to exercises

The target of traditional exercises is to help understanding theory, applying it to simple cases and providing a feedback to the teacher through the delivery of the solutions. In our system exercises are presented as HTML pages, containing text and figures of the assignments. The role of Deeds is to allow students to work out a solution, or to check its correctness, when obtained manually, and to provide graphical tools for editing the web page containing their reports. When learners are satisfied with their work they use Deeds to deliver the reports through the Internet.

The use of Deeds implies also a different approach to the structure of the exercises. In fact, with the simulator, students are naturally tempted to skip manual analysis. Exercises, therefore, must be targeted more to the real understanding of the issues than to the execution of repetitive tasks.

3.3. Design of digital systems

The development of a digital design project is the field where Deeds can fully be exploited. In fact, the interactive logic simulator, the finite state machine module and the microcomputer board emulator can work simultaneously in the simulation of a system where standard digital components can be controlled by state machines and/or a microcomputer board, as it is the case in contemporary digital design. Obviously, the modules can be used independently, to test separately the system parts.

The approach is meant to replicate the features of a professional environment, within the guidelines suggested by the educational purpose of the project.

Students use Deeds to download the assignment from a web page. The assignment consists of a functional description and a set of specification of the system that students must design. Project development phases are guided by help and instructions supplied through the Assistant browser, even if such instructions, in this case, are at higher level and the use of the simulation tools is less guided and left more to the user initiative. The use of Deeds in the development of a project find is, obviously, highly synergic with the tools provided by the NetPro systems, described above. NetPro emphasise the cooperative nature of project work, strengthens the skills needed in project management, project documentation (definition, plan, report), and project communication (internal, external, face-to-face, Internet-mediated).

4. AN EXAMPLE OF A LABORATORY SESSION

In Fig. 5 a list of laboratory assignments is opened in the Deeds main browser. The student executes the assignment #8.1: “Design of a synchronous mod-5 up/down counter”. With a click of the user on the link, the specific assignment will be opened in the Assistant (partial view in Fig. 6). The assignment asks the user to design a synchronous mod-5 up/down counter, using the Finite State Machine Simulator.

In the laboratory assignment is explained that the counter should generate a numerical sequence on the outputs QC, QB and QA, depending from the line input EN and DIR. The counter is synchronous with the clock CK and it is initialized by an asynchronous Reset input. In particular, the input DIR defines the count direction (up or down), and the input EN enables the count operation, that will take place on every clock positive edge.

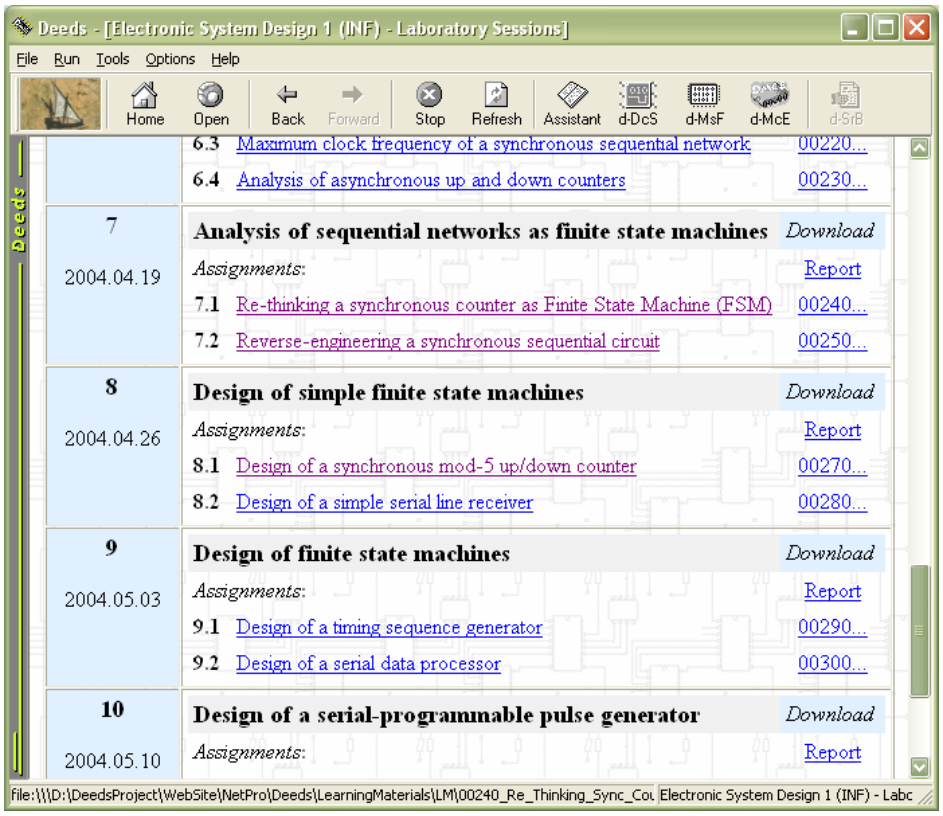

Fig.5: The Deeds Main Browser, here, shows a list of laboratory assignments.

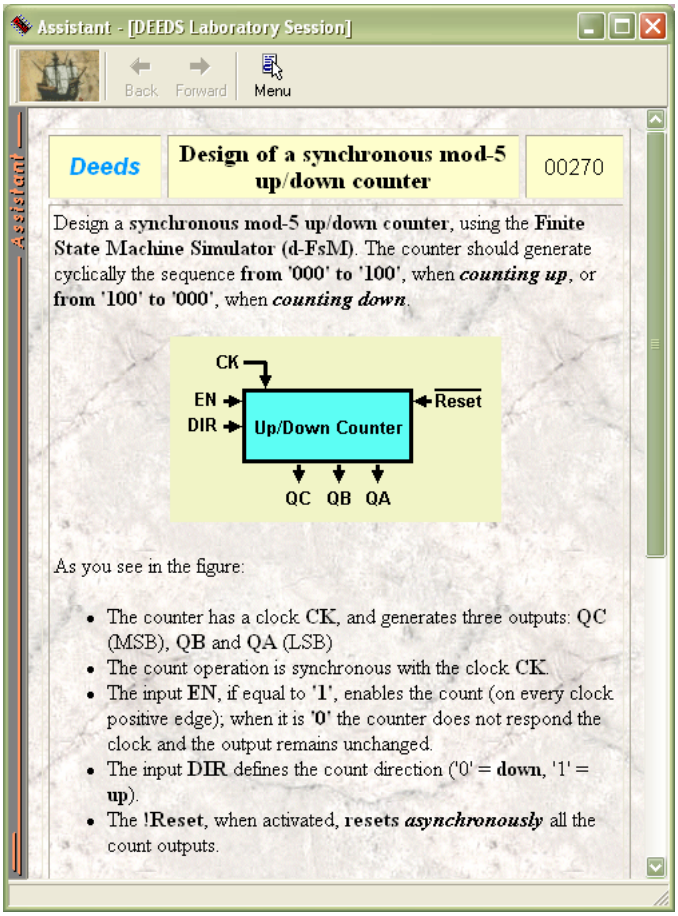

Fig.6: The specific laboratory assignment, discussed in the paper, opened in the Assistant browser.

The assignment continues with a suggestion: to download an ASM diagram template, to be guided toward the solution. If the student use this option, he or she could concentrate better on the argument, instead of build from scratch the solution, bothering with the simulator details and spending time in less useful and distracting tasks. The option is not mandatory, however, and the student can freely activate the simulator without using the template.

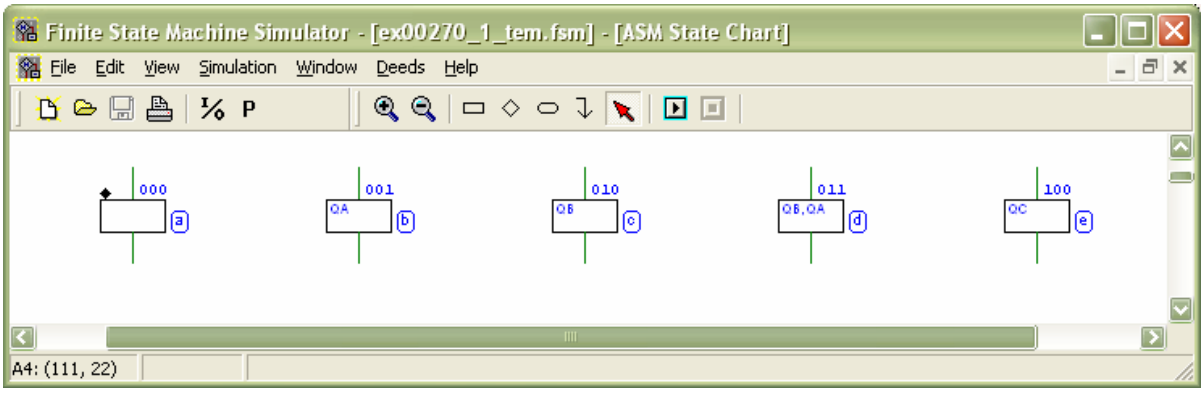

Fig.7: The ASM template of the solution, downloaded in the d-FsM.

To download the template, it is necessary only a simple click on the link in the text. The d-FsM will be activated, and the file downloaded from the web site, automatically. In Fig. 7 you see the suggested template, as downloaded in the simulator. In the template, as the text of the assignment explains, the student will find some important definition already set: the state variables X,Y,Z, the outputs QC, QB, QA and the inputs DIR and EN. The necessary five state blocks are already drawn. Actually, all the states properties have been pre-defined in the template. The user can modify this properties opening the Property Window (Fig. 8). This can be left aside to the editor, during the operations.

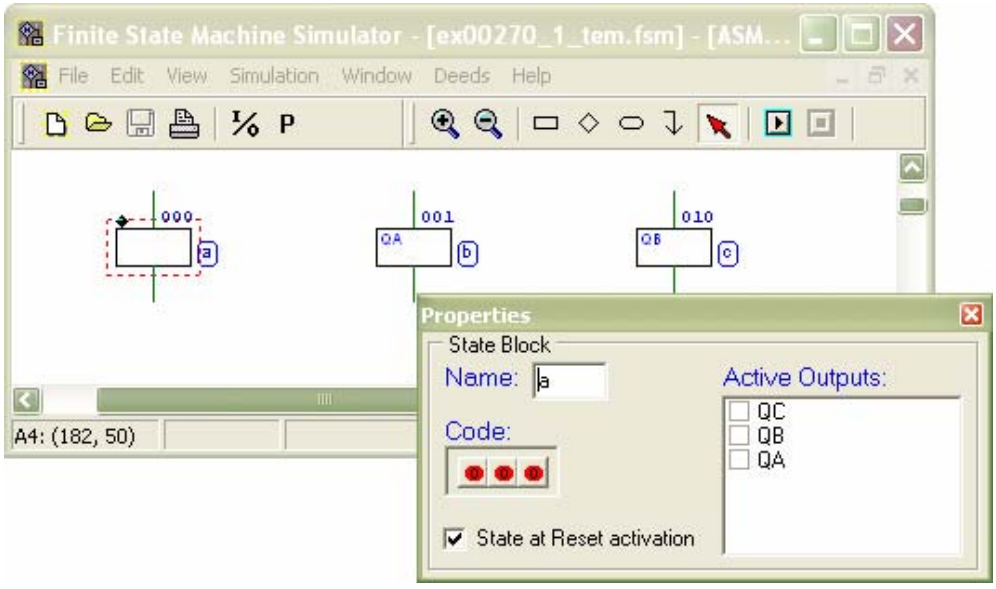

Fig.8: The property window, displaying the properties of the ‘a’ state.

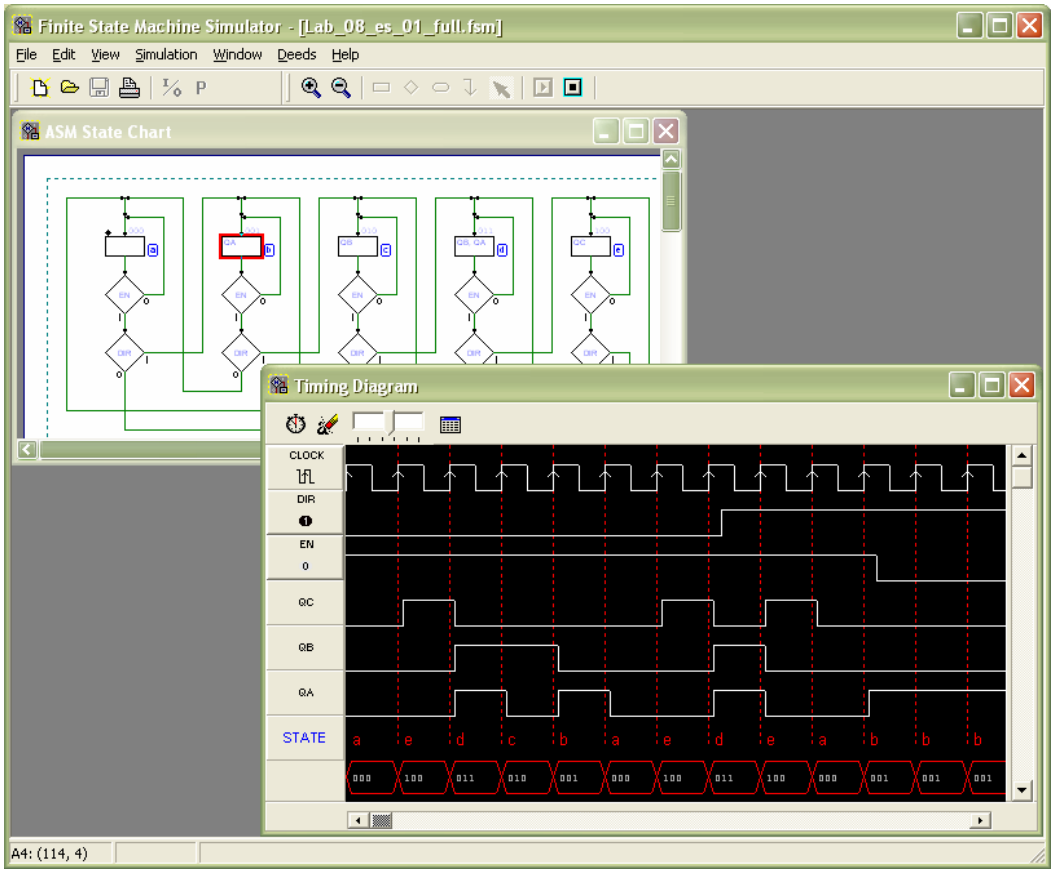

Fig.9: The completed ASM diagram, opened in the d-FsM, and its functional timing simulation.

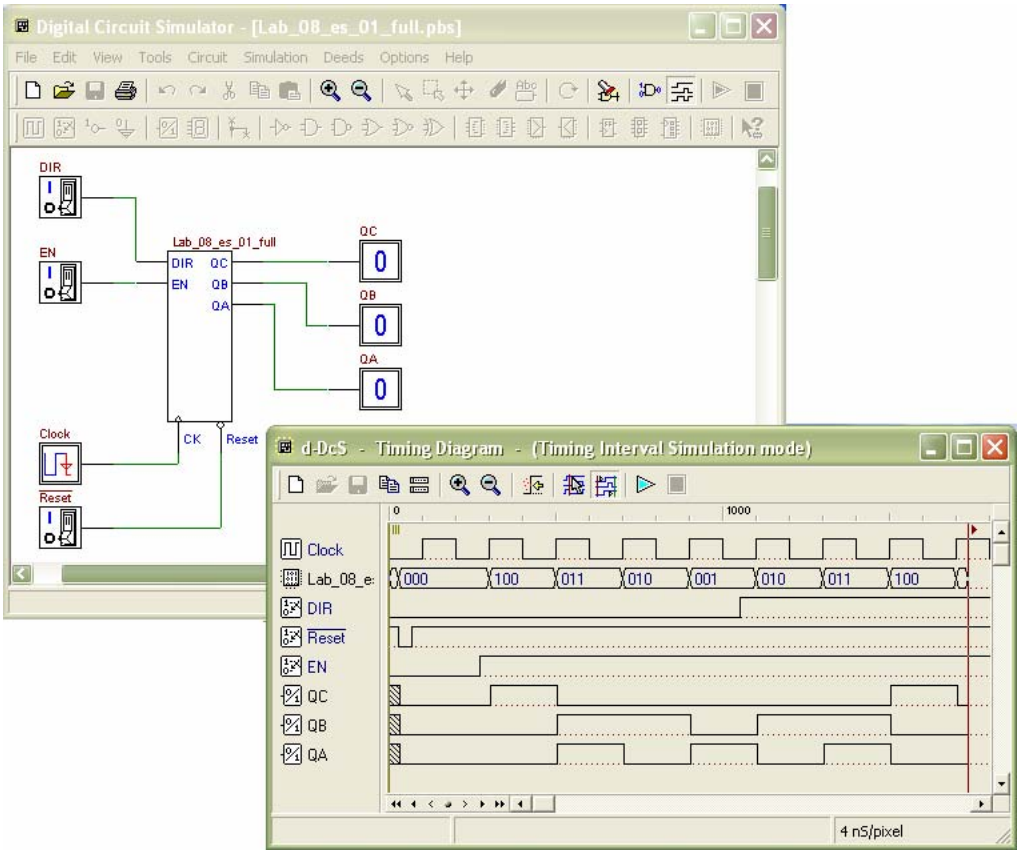

Fig.10: The completed d-DcS schematic, and the timing simulation of the component, in the d-DcS.

The user is asked to complete the ASM diagram and, using the timing simulation integrated in the d-FsM, to verify the correct sequence of output values and state codes. The user will start drawing, adding path lines and diamonds, as required by the requested functionality. Once the student have finished the design, the next step required is to verify the behaviour of the counter with the timing simulator of the d-FsM itself (Fig. 9).

When the user clicks on the 'Clock' button, the internal simulator evaluates next state and outputs (according to the current input values) and displays the results on the time diagram. At the same time, in the editor window, the corresponding new state is highlighted (with a coloured frame around it, see Fig. 9). This is an important feature, because a major difficulty, for a beginner, is to understand the correspondence between states and events time sequence.

Finally, when the behaviour of the component satisfies all the required specifications, the component could be imported in the d-DcS. Also a simple d-DcS schematic template is provided, to speed up the operations; it can be easy downloaded and opened in the d-DcS with a click on the hyperlink in the text. Once completed the schematic, the simulation of the counter could be repeated in the d-DcS timing simulator (Fig. 10).

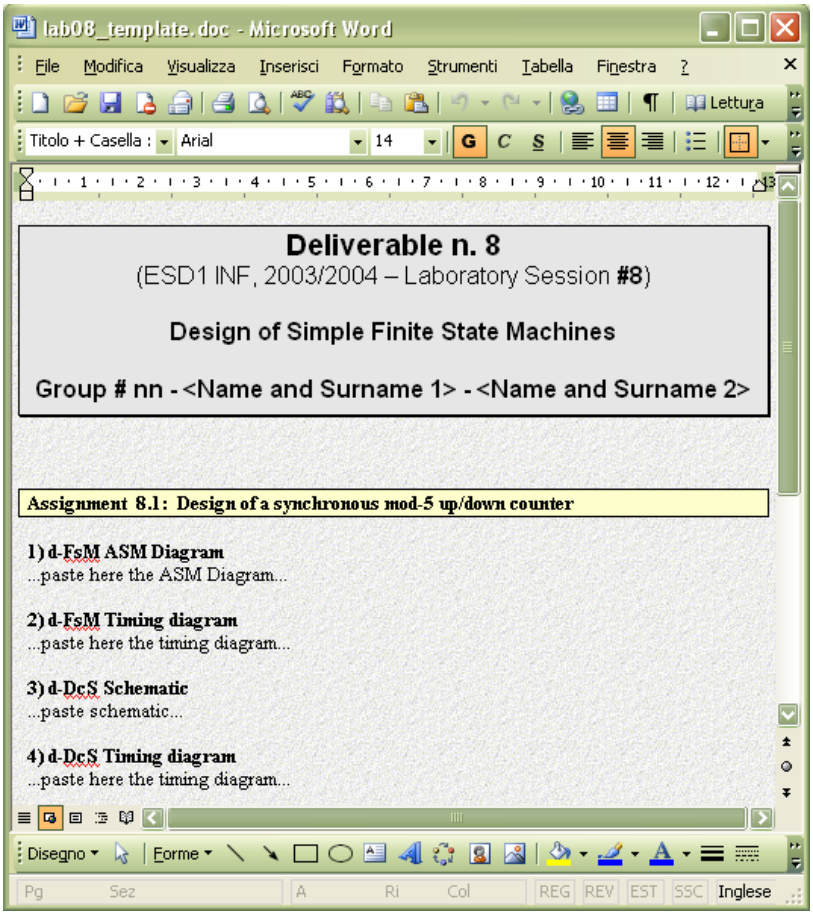

Fig.11: The report template for the laboratory assignment.

As in the example related to the d-DcS, at this point the student will compile and deliver a report about its work: in the assignments page, on the right, a link is set to download a report template file (Fig. 11).

5. THE NETPRO PROJECT

NetPro [3, 4], a European project of the Leonardo DaVinci program, develops project-based learning through Internet. It has created models, tools and services to facilitate communication and collaboration between distant students, and to manage access and control of project deliverables. Deeds has been developed as common simulation tool to be shared among different institutions running courses on Digital Design.

We test NetPro methodologies and tools by running pilot projects: four streams of pilots have been run during the project life and one of them [5, 6] is targeted to the field of electronics.

An important characteristic of the pilot courses is that project groups can be distributed over different academic institutions and countries. A pilot course may have teams from more than one institution and more than one nation while teams themselves could be inter-institutional and international. Joint working is possible if teams use the same language (all the components of our pilots, including student deliverables and communication, are in English) and if the classes involved study the same topic at the same time of year. Another extremely important condition for the sharing of courses is the availability of common tools, both for teachers and students, especially in the field of digital electronics, where circuit simulation plays a decisive role in the learning process.

Deeds fulfils that need, by providing not only the simulators, described later in the paper, but also a set of tools facilitating the preparation of learning materials for teachers and reports from students. All documents produced are available as web sites for on-line fruition or as downloadable files.

6. CONCLUSIONS

The use of software simulators, combined with Internet technologies, opens new perspectives for the development of learning environments for scientific and technologic disciplines [7]. Deeds places itself midway between a professional simulator and a pure pedagogical tool., filling the gap that students experience when confronting themselves with the former ones. The integration of the three simulators represents a novel feature in a pedagogical tool, in line with the current approaches of digital system design.

The custom development of Deeds is a huge investment of resources, whose convenience may be doubted by many, in view of the fact that professional CAD systems are currently offered for free to educational institutions by the manufacturers of programmable logic devices.

After several years of development work and pedagogical testing [8, 9, 10, 11], we can say now that our enterprise had and has a sense. Hundred of students are using Deeds in our lab and more students in NetPro partner institutions are sharing it. The development team is taking advantage of the vast experimentation to improve the functionalities and add new pedagogical features. Further information on Deeds is available at its website [12].

APPENDIX 1: TECHNICAL NOTES

The Deeds environment runs under Microsoft Windows (Win32 API); it requires about 10 Mbytes of free disk space, and at least 128Mbytes of Ram.

The Deeds has be written in Object Pascal language, using the Borland Delphi package and its standard VCL library. It uses also the Web-Browser Microsoft component. Its source code is composed of about 150 thousand lines (besides of 'forms' and other resources), and the project occupies about 20 Mbytes (of source codes, forms and resources, excluding VCL libraries and executables).

REFERENCES

[1] L. Anido, M. Llamas, and M.J. Fernández, " Internet-based Learning by Doing," IEEE Transactions on Education, Vol. 44, No. 2, abstract p. 193, full paper on CD, 2001.

[2] A. Bagnasco, M. Chirico, D. Ponta, A. M. Scapolla, "Laboratories in Internet to improve e-learning," in Proc Eunite 2002, European Symposium on Intelligent Technologies, Hybrid Systems and their implementation on Smart Adaptive Systems, Albufeira, Algarve, Portugal, ISBN 3-89653-919-1, pp. 685-689.

[2] NetPro, Project Web Site: http://netpro.evtek.fi

[3] H. Markkanen, G. Donzellini, D. Ponta, "NetPro: methodologies and tools for Project Based Learning in Internet," in Proc. ED-MEDIA 2001, World Conference on Educational Multimedia, Hypermedia & Telecommunications, ISBN 1-880094-42-8, pp. 1230-1235.

[5] NetPro, Electronic Engineering Pilot Courses: http://www.esng.dibe.unige.it/netpro/

[6] D. Ponta, G. Donzellini, H. Markkanen, "Electronic system design: an experiment of project-based learning on network," European Journal of Engineering Education, Vol. 26, No. 4, pp. 375- 390, 2001.

[7] M. Llamas, L. Anido and M.J. Fernández, "Simulators over the Network," IEEE Transactions on Education, Vol. 44, No. 2, abstract p. 212, full paper on CD, 2001.

[8] D. Ponta, P. Aiello, F. De Vincenzi, G. Donzellini, "A Multilevel Interactive Digital Circuit Simulator," in Proc ICTE'97, International Conference on Technology and Education, Oslo, Norway, pp. 276-278.

[9] A. Bovone, G. Donzellini, U. Pedrotti, D. Ponta, "A functional Finite State Machine simulator," in Proc CAEE'99, 5th International Conference on Computer Aided Engineering Education, Technical University of Sofia, Sofia, Bulgaria, 1999, pp. 85-90.

[10] G.Da Bormida, D.Ponta, G.Donzellini, "Methodologies and Tools for Learning Digital Electronics," IEEE Transactions on Education, Vol. 40, No. 4, abstract p. 294, full paper on CD, 1997.

[11] G. Donzellini, D. Ponta, "DEEDS: an Integrated Learning Environment for Electronics", in Proc. of SAAEI 2003, Seminario Anual de Automática, Electrónica Industrial e Instrumentación Decima Edición - Vigo (Spain), September 10-12, 2003, ISBN 84-688-3055-6, full paper on CD.

[12] Deeds Web Site: http://www.esng.dibe.unige.it/netpro/Deeds/.