Eurographics Italian Chapter Conference (2007)
Raffaele De Amicis and Giuseppe Conti (Editors)

Use of a 3D Graphical User Interface in Microelectronics Learning and Simulation of an Industrial Application

E. Bilotta[1], P. Pantano[1], S. Rinaudo[2], R. Servidio[1] and V. Talarico[3]

[1]Evolutionary Systems Group, Università della Calabria, Rende, Italy
[2]ST Microelectronics, Catania, Italy
[3]Mathematik und Naturwissenschaften Lehrstuhl für Angewandte Mathematik/Numerik, Bergische Universität, Wuppertal, Germany

Abstract

The COMSON project (COupled Multiscale Simulation and Optimization in Nanoelectronics) is a Marie Curie Research Training Network (RTN) supported by the European Commission within the 6th Framework Research Programme of the EU. One of the outcomes of the COMSON project is the e-learning system linked to the Demonstrator Platform (DP). This framework is intended for researches in the Microelectronics, Physics and Applied Mathematics fields. It is also intended for industries and expert technical personnel. The main aim of the project is to provide an effective training to students and young researchers with a hands-on experience with the DP, which comprises coupled simulation of devices, interconnects, circuits, electromagnetic (EM) fields and thermal effects in one single framework. Furthermore, the project has a Virtual Working Place system in which it is possible to share experiments and results among the partners. As this domain of knowledge is very complex - there are strong requirements in Industry in order to train young researchers and improve the know-how in a process of continuous education in these topics - the approach we propose overcomes the traditional difficulties of delivering contents, providing the user with a transparent interface, which hides the mathematical technalities. The user has only to follow a procedure by using a visual, tactile and explorative interface. It allows a graphical access to a preset library of models in the form of equations and algorithms for model manipulation. Furthermore, in this approach the researcher can define new models and algorithms by means of a constructivist metaphor. Finally, this method permits an easy control of the DP simulation function, together with a complete monitoring and data analysis facility.

Categories and Subject Descriptors (according to ACM CCS): I.6.2 [Simulation and Modeling]: Simulation Languages

1. Introduction

The COMSON project (**CO**upled **M**ultiscale **S**imulation and **O**ptimization in **N**anoelectronics), begun in 2005, and has been financed for four years by the European Commission. Its aims at building an experimental software for coupled simulation of devices, interconnects, circuits, EM fields and thermal effects (one single framework for simulation tool, optimisation in compound design space) together with an e-learning system, experimentation and collaboration platform. COMSON merges the know-how of the three major European semiconductor industries, with the combined expertise of five specialized university groups. One of main focuses of COMSON is to provide an e-learning platform comprising learning management system (LMS) especially devoted to training in micro and nanoelectronics simulation[ABPT]. The LMS has to provide facilities to interface a remote simulator. Furthermore the e-learning platform integrates many different technologies (HTML, SCORM, FLASH, PDF, media streaming etc.) to allow flexibility in the definition, composition and interconnection of learning contents. Some of those technologies will enable multiple interaction modalities to better suit the need of an heterogeneous audience ranging from students to researchers, from to institutional partners to third parties. The e-learning platform assures:

- high scientific quality of the course

- mathematics dedicated to industrial needs
- interdisciplinary and intersectorial education
- career training
- project and communication
- intensive transfer of knowledge (ToK)
- timely communication

The LMS allows the management of courses, students, didactical modules, real-time and differed learning. Tools are available to teachers and devoted to authoring of lessons and assessment tests.

This authoring environments are projected to include principal aspects of the constructivist learning theory [MRK04], in particular the possibilities to visualize (with animation) and to manipulate interactively, learning objects or metaphors of learning objects. The e-learning platform provide contents based upon the Sharable Content Object Reference Model (SCORM) standard. Moreover, the SCORM support is integrated with distributed technologies in order to develop a complete learning system. The distributed approach (e.g. Demonstrator Platform architecture and COMSON Virtual Working Place) provides the capability for applications to run remotely. Intense research activity is ongoing on e-learning technologies especially focusing on accessibility, interoperability, durability, and reusability of components [PMM02]. Applying Web Service Technologies to a SCORM compatible LMS simplifies the implementation and maintenance of the LMS and gives to web service consumers more choice in finding the services they require [CCYY04][CCYY03].

The COMSON creates a framework in which it is possible to develop a Collaborative Working Environment (CWE) based on simulation software, in which it is possible to test mathematical methods and approaches, in order to assess whether they are capable of addressing the industry's problems and to adequately educate young researchers by obtaining immediate hands-on experience for state-of-the-art problems.

The experimental simulation software used to test models and simulation algorithms is called Demonstrator Platform (DP). A drawing of the DP architecture is shown in 1. This software has been conceived as an open framework where users can define mathematical models describing the systems behavior in physical terms and boundary conditions (e.g.: using specializations of Maxwell Equations), define different algorithms to extract numerical solutions and test the predictivity of models along with the algorithm's performance.

Furthermore, the DP architecture has been designed to be open in order to allow interoperability, through a suitable set of software plug-ins, with domain specific software applications (e.g.: Cadence suite and the like). The interoperability effort has been taken to let the DP benefit from third party software libraries of functions, models and algorithms and to provide a common framework suited for coupled simulation and evaluation.

Coupled simulation allows the investigation of physical phenomena that happen when there is a significant interaction between the mechanical, electrical, magnetical and thermal behaviors of a device. Not taking those phenomena into consideration when designing and simulating a device may lead to failures in production. Coupled simulation technologies improve simulation accuracy and some of the applications where multi-domain analysis makes a comprehensive reliability study feasible and practical.

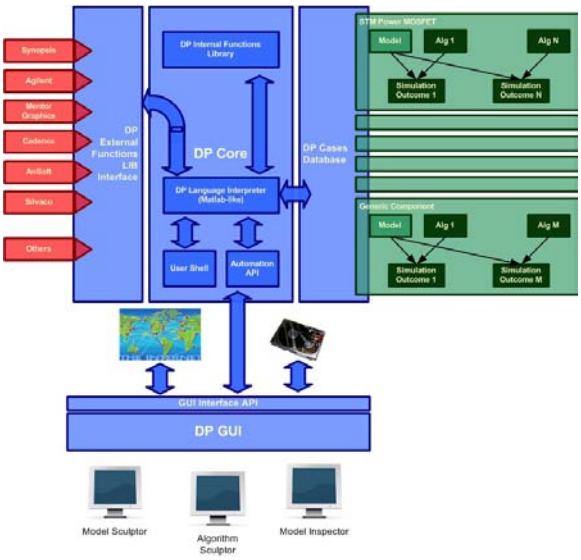

Figure 1: *The DP architecture.*

Another important aspect of the DP architecture is the possibility to integrate vertically the electronic design process thus allowing the implementation of and EDA (Electronics Design Automation) system.

Finally, the DP allows to simulate models' behavior at different scales using different user-defined and pre-packaged simulation algorithms.

The DP user interface is mainly base on the Octave open source numerical analysis software [Eat97]. This means that the user interface is text based and it accepts commands and scripts in a Matlab-like programming language.

The DP is certainly a fundamental asset to pursuit the objectives of the COMSON when the research focus is only on technology. But if the attention is shifted on how to enable people and organization to practically use those technologies, it can be easily understood that technology by itself is not sufficient for problems domain definition, understanding of problems and solutions, implementation of solutions ad adoption into production environments. Thus it becomes clear, by focusing on end users, that introduction, experimentation, evaluation and adoption of a new technology into a production or training context has certainly to do with the scientific quality of research from which the technology had been derived, but has a lot more to do with the implementation of a correct, complete and effective Transfer of Knowledge (ToK) process.

In fact, design, modelling and simulation of an electronic devices is a complex activity involving multidisciplinary teams of designers. Design teams often have to sum up competences from the electronics, mathematics, physics fields and are made of people living and working at different sites. Starting from this consideration, it has been designed within the COMSON collaborative work environment whose objective is to provide an effective and complete infrastructure devoted to knowledge organization and sharing, experimentation and learning [ABGP06].

Before introducing the VWP infrastructure it is necessary to point out who are the users the COMSON project is addressing its research to. Two different classes of user have been identified that will use the COMSON technologies: users with experience in the fields that are background prerequisites to take advantage of the DP and the theoretical scaffolding that surrounds it; users without any experience both in the theoretical framework and scientific background and/or without experience of device design, simulation and production. The first group mainly comprises user with strong academic background and specific insight on the domain and class of problems addressed by the COMSON research effort. People from this group can belong to industry or academy and can be moved by practical or theoretical need to experiment within the VWP infrastructure. The second group of user is made of undergraduate students, PhD students, young researchers and industry trainees with some basic academic background but almost no understanding of the complex theories and techniques made available through the COMSON project. Thus this group needs to build from the ground up expertise and proficiency with the DP in general and with coupled multiscale design and simulation specifically.

With this groups definition in mind, it has been thought that the COMSON collaborative work environment should have made available a set of TOK tools that, by means of multimodal interfaces expressely designed to allow users to explore knowledge at different degrees of complexity e completeness, allow users to interact with the system in a natural way removing as far as possible all of the complexity in learning due to interfaces and implementing familiar exploration, manipulation and collaboration metaphors.

The COMSON collaborative work environment has been called Virtual Work Place (VWP). It comprises three interconnected facilities, each devoted to a different aspect of the collaboration and TOK processes. A schematic representation of the VWP organisation in shown in fig. 2. The first component of the VWP is the virtual museum. This component has been designed to represent a three dimensional environment inside of which the user controls a graphical avatar. Each room in the museum has a theme related to microelectronics. There can be found rooms for historical figures, discoveries, technologies and so on. The user is free to move his or her avatar exploring the museum and moving from one ambient to the other. The museum is provided with intelligent 3d agents that can be used to gather informations by request, get suggestions on exploration paths and have support in accessing other services. An example of the virtual museum navigation environment in shown in 3. Furthermore, the virtual museum is a multiuser environment. This means that user can meet and communicate within the museum. The virtual museum has been developed using Java 3D.

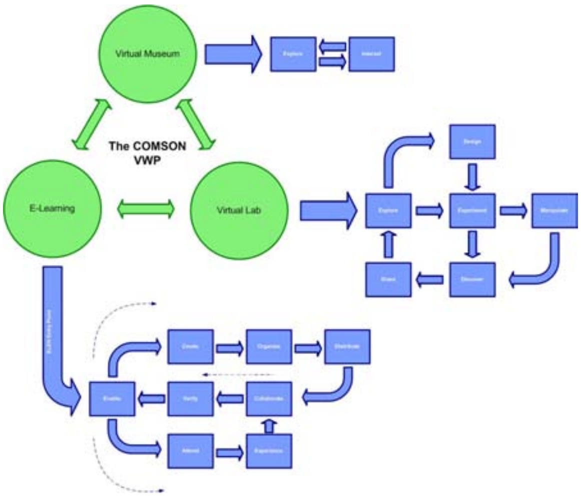

Figure 2: *The VWP architecture.*

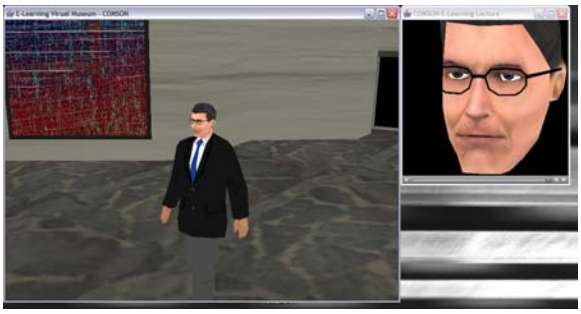

Figure 3: *The Virtual Museum.*

The second component of the VWP is the e-learning environment. This environment can be accessed through an entry point provided at the museum and allows the user to explore learning materials, courses, assessment tests, experimentation facilities, examples etc. The e-learning environment can accept and provide learning material in a large number of formats and supports the publication of courses based on the SCORM standard [Dod06].

When a user of the e-learning environment accesses the experimentation platform, the third VWP component comes into play; it is the Virtual Lab. This component's user interface is the three dimensional reproduction of an electronics laboratory populated with instruments and electronic components. By interacting through their avatars, users can experiment with instruments and electronic components both in an autonomous way and being guided by intelligent agents. The Virtual Lab allows to access the DP through a graphical user interface (GUI) which uses a visual programming language (VPL). A VPL is any programming language that lets users specify programs by manipulating program elements graphically rather than by specifying them textually. A VPL allows programming with visual expressions, spatial arrangements of text and graphic symbols [EE*99]. This VPL has been especially designed to let the user interaction with the DP be easy, intuitive and immediate without the need to access any command line interface.

The approach of hiding all the complexity and details of work environment or software application behind a well known and well metaphorized user interface, the GUI, has proven to be practical and effective in reducing learning curves and acceptance of new software applications. Furthermore this approach has been successful when adopted into collaboration contexts [BID98].

In the following paper will be presented the design for the DP visual interface along with an industrial case study. First it will be given a description of the DP GUI, its integration into the VWP, its components and their functionalities. For each component of the DP GUI a simple usage example will be presented. Then an industrial application case will be discussed into detail presenting a theorethical and techonological backgound. An overview of the classical approaches to the application will be given and, finally, it will be shownhow the DP GUI can be used for such application.

2. The DP's GUI

The DP visual interface (VI) is a Java application tha can be run either into a web browser or as a stand alone module. This makes it possible to integrate the DP VI into the VWP at the web level letting it be accessed both form the e-learning environment and from the Virtual Lab.

Furthermore, referring to 2 showing the VWP architecture, the DP VI is therein integrated at the Experiment, Explore and Manipulate level of the Virtual Lab sub architecture and at the Experience level in the e-learning environment. This means that user exploring both the virtual lab and the e-learning contents can gain direct access to the DP VI. The access grants the user with the possibility of conducting real experiments on the real simulator.

Another aspect of the GUI is that it has been designed as a client interface to a DP server. This permits the use of the GUI for design and manipulation purposes on local models even if there is no connection to the DP. The architecture of the GUI is shown in fig. 4.

When connected to a DP server, the GUI allows the exploration of the remote database containing models and algorithms and the use of library functions imported from third party software.

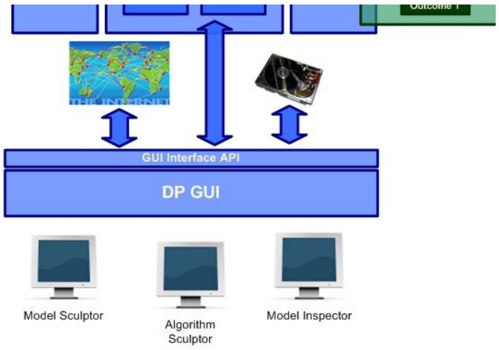

Figure 4: *The DP GUI architecture.*

The GUI offers three work environments: the model sculptor, the algorithm sculptor and the model inspector. The first two environments are 3D authoring tools for, respectively, designing and manipulating mathematical models (equations) representing the devices and for designing and manipulating algorithms providing numerical solutions to those equations.

The third environment is a tool for inspecting the value of variables in the model during simulation time.

2.1. The Model Sculptor

The Model Sculptor is 3D authoring environment that allows the user to create mathematical models by defining equations through manipulation of simple 3D objects representing boxes. Choosing the right semantics and associating to different boxes different meanings, it has been possible to describe complex mathematical models by combining those boxes spatially.

The semantics of boxes was inspired by the reverse polish notation (postfix). An example of this concept is given below. The expression

$$a+b$$

in common notation traslates into

$$+\; a\; b$$

in RPN notation.

A possible 3D representation of this RPN expression, using the boxes semantics, is shown in figg. 5 e 6 .

In fig. 5 and 6 the red box formalism is used to indicate operators while the blue box formalism is used for literals.

This kind of representation is a tree hierarchical one. For complex models, the Model Sculptor allows the user to define the degree of complexity of the representation by choosing the depth of the exploded graph. The Model Sculptor is provided with a fully featured environment for composing models using a set of operators (e.g.: sum, subtraction,

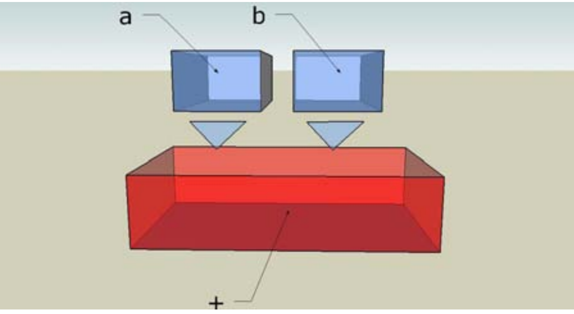

Figure 5: *An exploded representation of the expression $a+b$ composed according to the graphical formalism provided by the Model Sculptor.*

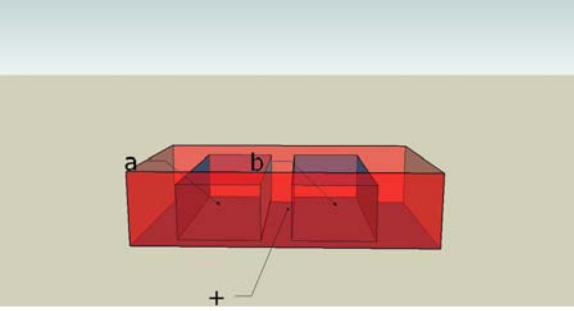

Figure 6: *A compact representation of the expression $a+b$ composed according to the graphical formalism provided by the Model Sculptor.*

multiplication, division, square root, exponential etc) that are binary and unary. Moreover it provides also some predetermined useful functions such as sinusoidal, logarithmic and exponential useful for defining input to models.

The Model Sculptor stores the representation of mathematical models using the MathML language. MathML is an XML specialization proposed by the W3C for the treatment, interchange and representation of mathematical formulas [ABC*03]. The MathML specification has a standard for the graphical representation of mathematical equations which is commonly referred to as the presentation component; another component of the MathML specification, called content, is used to preserve information about the semantics of mathematical expression and is commonly used form machine interpretation.

By exploiting the content component of the MathML specification, the Model Sculptor can interoperate with the DP translating visual representation of hierarchies of 3d objects with mathematical semantics into MathML content. This transformation guarantees the preservation of semantics against the mere use of the presentation component of MathML. Subsequently the MathML content can be translated into a sub set of the Octave language used to represent mathematical expressions for the elaboration into the DP.

Conversely, the Model Sculptor can import and translate into a graphical representation expressions written using Octave's syntax.

2.2. The Algorithm Sculptor

The Algorithm Sculptor is used to design algorithms used to manipulate mathematical model in order to find numerical solutions to them. The environment has been designed in order to be able to associate different algorithms to the same model so as to make it possible to compare algorithms performances in terms of computational complexity, and thus execution time, and accuracy of the results.

As it happens form the models composed in the Model Sculptor, algorithms can be designed and manipulated into the Algorithm Sculptor environment by the use of 3D representations of data transformations and data transfers.

The Algorithm Sculptor provides a 3D environment for designing algorithms graphically by combining simple elements (spheres and sticks) into complex computational architectures. Each sphere is used to represent simple data transformations or functions. Each stick is oriented and connects two spheres signifying both an execution order constraint and a flow of information between two spheres.

By providing, through the DP, access to a whole library of algorithms and numerical methods it become simple to import solvers for the equations, connect variables to function inputs, define parameters and visualize the results.

This formalism makes easy to combine also different algorithms and experiment with the results.

Fig. 7 shows a common first order differential model while fig. 8 shows a example of a compact representation a solution algorithm using the Runge-Kutta method [Zwi00]. In fig. 8 can be seen three different types of elements: spheres, oriented sticks (or cones), and a diamond. The diamond represents a condition in the algorithm. Every sphere can have one or more labels. The black labels are the names of the functions (or transformations) carried out by the sphere. The green labels are place holders indicating that a quantity or

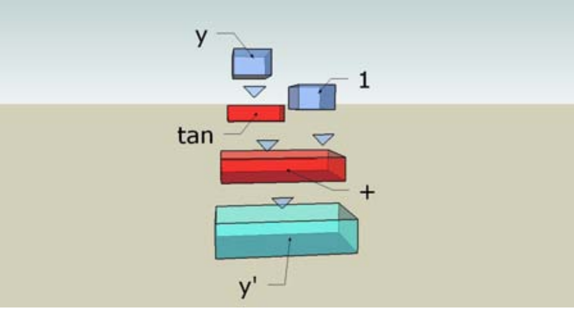

Figure 7: *The model for the differential equation $y^I = tan(y)+1$.*

variable being handled at that specific sphere has to be monitored (its value sampled) every time the sphere is active, i.e. its code executed. The sampled values are passed to the Model Inspector for visualization. Sphere can be exploded to visualize the implementation details of the contained functions thus providing a hierarchical view of the algorithm mechanics.

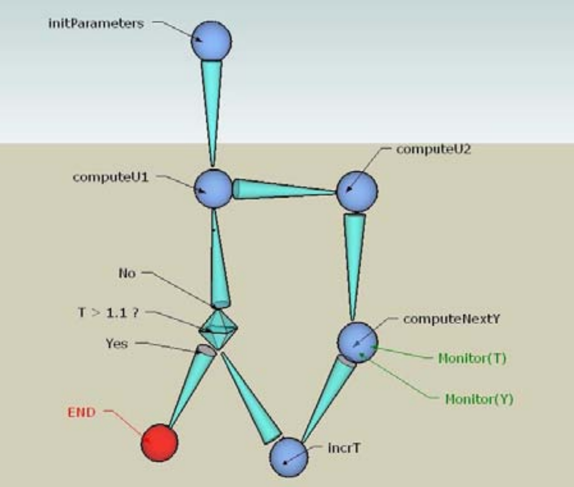

Figure 8: *Algorithm implementing the two stage Runge-Kutta explicit method.*

Algorithms can be executed step by step, to verify the operations, or continuously to quickly get the results.

2.3. The Model Inspector

A typical visualization of the Model Inspector is shown in fig. 9. Every variable defined into a model or algorithm can be sampled and drawn into an inspector plot. The results of every execution can be stored along with the model and algorithms. The stored data can be loaded later by the model inspector in order to compare results from different simulations with different parameters sets.

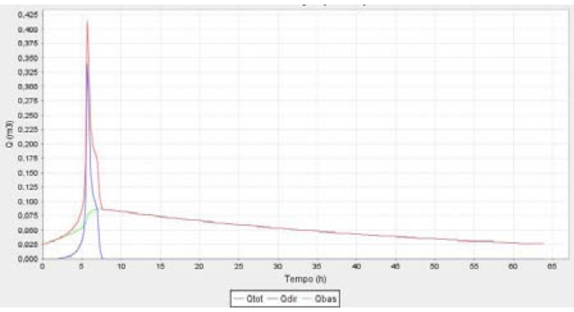

Figure 9: *Variables plotted against time in the Model Inspector.*

It is also possible to load meshes representing devices under inspection and associate spatially result sets to those meshes to generate animations that can be saved in various video formats or exported into 2D raster images.

3. Industrial case application

The design of integrated insulated gate power devices (PowerMosfet, Power bipolar and IGBTs) has to deal with issues due to electrical and thermal effects that cannot be simulated using conventional circuit simulation software. It is thus required to take some different approach in defining and simulating this kind of models.

This class of devices can be discretized by a matrix structure where each cell is an elementary power device [BGA*]. Classic concentrated models do not take into account this matrix structure therefore they cannot predict certain device behaviors at given spots on the matrix because they assume that upon switching on and off the devices every single cell receives the signal at the same time and reacts simultaneously with the others.

The behaviour of the real devices differs from the classic simulated one because the signal propagates ad different speeds when travelling into the substrate instead of the gate contact. A correct model has to take into account this propagation delay. Furthermore, upon device switching, classic model fail in determining the device beaviour because they do not take into account phenomena such as current distribution and other switching anomalies. Those anomalies can cause currents to concentrate in well determined areas of the device and, if the design is not carried out taking into account those concentrations, the device may burn and thus the production may fail [BGBR06][BGBR05].

In the rest of this section a method will be described for modeling regime behavior and it will be implemented and tested using the DP VI.

As mentioned before, the power device can be described using a matrix of elementary power devices. Those devices are interconnected to the signal source in parallel using a line connected to the device gate. Every lines, called "‘fingers"’, depart from the gate and bring the signal to parallel rows of cells. Cecause of this peculiar gemetric structure, it is possble to examine just one "‘finger"’ and then extend the results to the other "‘fingers"’.

Every finger can be modeled through an ideal monodimensional trasmission line characterized by a resistence per length unit R and a capacity per length unit C when inductive effects can be considered non influent. Holding this hypoteses true, the voltage $u(x,t)$ and current $i(x,t)$ along the line are determined by the following equations:

$$\frac{\partial u}{\partial t} = \frac{1}{RC}\frac{\partial^2 u}{\partial x^2} \tag{1}$$

$$\frac{\partial i}{\partial t} = \frac{1}{RC}\frac{\partial^2 i}{\partial x^2} \tag{2}$$

Refering to equation 1, the following intitial conditions are used to determine an analytical solution:

$$u(x,0) = 0 \; \forall x \geq 0, \tag{3}$$

and the boundary conditions

$$u(0,t) = \alpha(t) \;\; \lim_{x \to \infty} u(x,t) = 0 \; (t > 0) \tag{4}$$

The ideal line condition implies no reflection and a voltage that is 0 when $x \longrightarrow \infty$. It is also possible to consider a non ideal line, i.e. where a certain amount o reflection occurs. This reflection can be attenuated by adapting the impedance of the line. Using Laplace tranform, under the conditions 3 and 4, it can be determined the following analytical solution of the voltage equation 1:

$$u(x,t) = \frac{x}{2\sqrt{\pi \frac{1}{RC}}} \int_0^t \alpha(t-\tau) \frac{\exp - \frac{x^2}{4\frac{1}{RC}\tau}}{\tau\sqrt{t}} d\tau \tag{5}$$

However, this solution is impractical because, in general, given a waveform $\alpha(t)$, it is not possible to solve analytically the integral.

If the boundary condition is set to be constant over time (i.e.: $u(0,t) = u_0 =$ constant $\forall t > 0$) then it can be shown that the solution is:

$$u(x,t) = u_0 \mathrm{erfc}\left[\frac{x}{2\sqrt{\frac{1}{RC}t}}\right]. \tag{6}$$

When the boundary conditions are set to be oscillating, and the equation is solved for t suffciently far from t_0, it is possible to ignore any effect due to the transitory response of the system. In this case, if

$$u(0,t) = A\cos(\omega t) \tag{7}$$

the solution of equation 1 is

$$u(x,t) = A\exp\left(-\sqrt{\frac{\omega}{2\frac{1}{RC}}}x\right)\cos\left(-\sqrt{\frac{\omega}{2\frac{1}{RC}}} + \omega t\right). \tag{8}$$

This solution can be compared to that deriving from simutation obtained by both solving numerically equations 1 and 2 and by modeling the device elementary "‘finger"’ using a dicrete RC circuit, adapted to attenuate reflection.

In the first case an impedence adapted network of 1585 elementary RC cells has been simulated. The schematic for the network is shown in fig. 10. In the second case the model has been composed into the model sculptor and simulated by solving it using different numerical methods [Kev00] in order to test and compare their effectiveness against each other and against the dicrete circuit model.

DP model used for the simultaion is shown in fig. 11. In fig. 11 there is a new element: a yellow cylinder. This element is used to define an equality relation between to terms. In this model operators and functionals are red, literals and constants are light blue while variables (to be calculated) are light green.

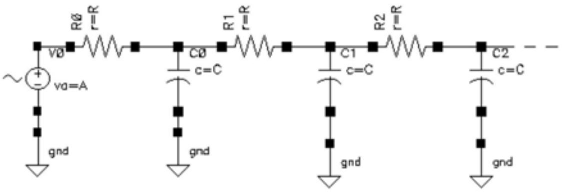

Figure 10: *An example of a discrete circuit model to simulate the integrated power device behaviour.*

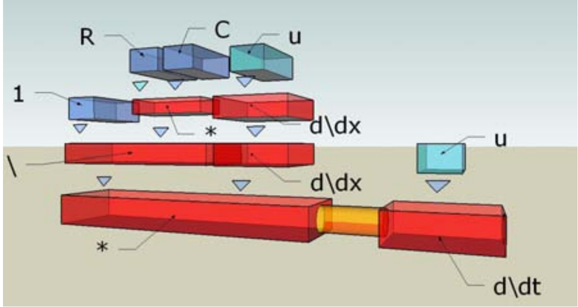

Figure 11: *The partial differential equation modeling the integrated power device composed into the DP TUI.*

4. Conclusions

Simulations carried out using the DP VI environment with different solution algorithms have shown that the solution of the equation modeling the physical device using different numerical methods has given results that are comparable, and in some cases better, than those obtained using the discrete circuit modelling and simulation method. However the quality of results varies significantly with the chosen numerical solution method and with the granularity of the method. An higher granularity is often paid in terms of an higher computational complexity.

Furthermore it has to be taken into consideration that in many case numerical methods for solving differential equations fail to converge to a valid solution.

However the manipulation and exploration of models and algorithms provided by the DP GUI has proven to be useful in context where learning by exploration and design by exploration are common approaches.

References

[ABC*03] Ausbrooks R., Buswell S., Carlisle D., Dalmas S., Devitt S., Diaz A., Froumentin M., Hunter R., Ion P., Kohlhase M., Miner R., Poppelier N., Smith B., Soiffer N., Sutor R., WATT S.: *Mathematical Markup Language (MathML) Version 2.0 (Second Edition)*. World Wide Web Consortium, http://www.w3.org/TR/MathML2/, 2003. 5

[ABGP06] ALÌ G., BILOTTA E., GABRIELE L., PANTANO P.: An e-learning platform for academy and industry networks. In *Pervasive Computing and Communications Workshops, 2006* (2006), pp. 169–178. 3

[ABPT] ALÌ G., BILOTTA E., PANTANO P., TALARICO V.: E-learning strategies in academia-industry knowledge exchange: the comson case.(submitted for publication). In *Pervasive Computing and Communications Workshops, 2007*. 1

[BGA*] BIONDI T., GRECO G., ALLIA M., LIOTTA S., BAZZANO G., RINAUDO S.: Distributed modeling of layout parasitics in large-area high-speed silicon power devices (to be published). 6

[BGBR05] BIONDI T., GRECO G., BAZZANO G., RINAUDO S.: Analysis of the internal current distribution in power mosfets operated at high switching frequency. In *Proc. Conf. of MSED 2005 - 15th Workshop on Modelling and Simulation of Electron Devices* (July 2005), pp. 169–178. 6

[BGBR06] BIONDI T., GRECO G., BAZZANO G., RINAUDO S.: Effect of layout parasitics on the current distribution of power mosfets operated at high switching frequency. *Journal of Computational Electronics 5*, 2-3 (July 2006), 149–153. 6

[BID98] BRAVE S., ISHII H., DAHLEY A.: Tangible interfaces for remote collaboration and communication. In *Computer Supported Cooperative Work* (1998), pp. 169–178. 4

[CCYY03] CHU C. P., CHANG C. P., YEH C. W., YEH Y. F.: Web-service oriented content providing for scorm compatible lms system. In *Proceedings of the 9th International Conference on Distributed Multimedia Systems (DMSŠ 2003)* (Miami, Florida, September 2003), IEEE Computer Society, pp. 220–225. 2

[CCYY04] CHU C. P., CHANG C. P., YEH C. W., YEH Y. F.: A web-service oriented framework for building scorm compatible learning management systems. In *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC)* (Las Vegas, Nevada, April 2004), IEEE Computer Society, pp. 156–161. 2

[Dod06] DODDS P.: *SCORM 2004 3rd edition, sharable content object reference manual*. Advanced Distributed Learning, http://www.adlnet.gov/downloads/files/311.cfm, 2006. 3

[Eat97] EATON J. W.: ** Octave: (octave). Interactive language for numerical computations*. The GNU Project, http://www.gnu.org/software/octave/doc/interpreter/, 1997. 2

[EE*99] EHRIG H., ENGELS G., , KREOWSKI H., ROZENBERG G.: *Handbook of Graph Grammars and Computing by Graph Transformation: Applications, Languages and Tools*. World Scientific Publishing Co. Ptc. Ltd., 1999. 4

[Kev00] KEVORKIAN J.: *Partial Differential Equations: Analytical Solution Techniques, 2nd ed.* Springer-Verlag, 2000. 7

[MRK04] MORRISON G. R., ROSS S. M., KEMP S. E.: *Designing effective instruction*. Wiley, Hoboken, New Jersey, 2004. 2

[PMM02] PAULO P. F., MARIO R. F., MARTA M.: Building reliable web services composition. *Revised Papers from the NODe 2002 Web and Database 2593* (2002), 59–72. 2

[Zwi00] ZWILLINGER D.: *Handbook of Differential Equations, 3rd ed.* Springer-Verlag, 2000. 5