ERCIM “Alain Bensoussan” Fellowship Scientific Report

Fellow: **Marek Gayer**
Visited Location : **CNR Pisa, Italy**
Duration of Visit: **14 months**

I - Scientific activity

My scientific activity was directed towards the goals of simulation and visualization of nanometer scale devices, in the field Semiconductor Technology (TCAD). Activity included research, analysis of state of the art, and later also designing and implementing an open source software platform as a solution suitable for simulation and visualization of TCAD problems.

The initial phases of the research were focused on finding existing solutions and architecture parts, together with defining a strategy for fulfilling goals of our TCAD simulation and visualization project. Some general goals and features were evaluated and planned in detail, so we would be able to propose technologies for a realistic realization of a TCAD software project. This included also the possibility of running simulation modules on remote servers, whose execution flow would be controlled by a client computer (either using a desktop application or within a web browser), and perspectives for integration with our already existing or newly created simulation codes.

Another activity was my research of optimal mathematical and numerical methods, which could be used as the base and the solution core of these goals. In particular, I studied finite volume and finite element methods, together with appropriate existing software and tools that could help us in our research goal. I analyzed and tested finite element and finite volume meshers (namely Tetgen, Netgen, Gmsh), pre-processors and post-processors. I put emphasis on obtaining advanced means and tools for peak interactive visualization possibilities for pre-processing and post-processing of TCAD simulations. Some implementation of basic prototypes based on those tools, that were expected to become parts of the solution, were carried out. The prototypes were integrated in existing finite element software “Salome Platform”.

After this activity, I together with prof. Giuseppe Iannaccone finally decided to choose the finite element method (together with some of the carefully selected software tools) as the core method and the base in our TCAD project. My further effort then was solving the problem of optimal selection of all the available free or open source finite element numerical packages necessary to compose a software abstraction of governing TCAD equations. I selected a solution which clearly seemed most suitable and promising for our needs, finite element software “Fenics” with its C++ interface “Dolfin”.

After finishing of all the above-mentioned activities, I was consequently able to design and partially implement a new research platform (with the preliminary name “NanoFEM platform”) for TCAD simulations. This platform is composed of some of thoroughly tested finite element software components (Salome Platform, Fenics/Dolfin) as well as with our own C++ libraries that I developed, which is connecting these components and providing additional features for modules. These libraries allow powerful, easy and rapid development of such modules.

Due to the following two problems: complexity of installation as well as the requirement of portability of the solution on Windows, Linux and Mac, I proposed and tested an approach of infrastructure based on virtualization, in particular using VmWare tools. This way, not only separated modules can be launched on different computers, but also whole software platform can run on remote servers and can be accessed by a client application (using VmWare Server and VmWare server console, which can be obtained freely). With such solution, I was able to address both these problems.

The whole solution was tested on computing linear and non-linear Poisson equation on rather complex MOSFET (FinFET) structures and meshes (e.g. 15 groups and 4 materials and on meshes with hundreds of thousands finite elements). The architecture should allow rapid development of further modules.

The current state of the project offers a very solid foundation for further development and realization, under the condition of available funding for our TCAD project. The solution allows the computation modules to run both on the client computer and on the server, interaction and visualization is available on client side, and is possible to be available even in the web browser (although under the condition of fast internet connection).

II- Publication(s) during your fellowship

None

III -Attended Seminars, Workshops, and Conferences

None

IV – Research Exchange Programme (12 month scheme)

1. CNR - Roma, Opto & Nanoelectronics group (OLAB) with Associate Professor Aldo Di Carlo, University of Rome "Tor Vergata", via del Politecnico 1, 00133 Roma, Italy, 5.5. – 12.5.2008.

Research visit focused on exchanging experiences with software projects solving TCAD simulation using the finite element method. These were our NanoFEM platform, briefly described above, developed at University of Pisa and TiberCAD developed at University of Rome, Tor Vergata, available on www.tibercad.org . Some of the approaches that were on this workplace discussed and compared follows.

Geometry editing:
TiberCAD: It offers the geometry editor provided in Gmsh. It is a half-interactive graphical editor, with possibility to specify geometry in text editor in .geo format. It does not support Boolean operations and number of operations is rather limited.

NanoFEM: Comes with powerful, fully interactive and comfortable graphical editor using Salome, with wide range of operations. Furthermore, it seem to be considerably more stable then Gmsh. It is also possible to define geometry using Python scripting.

Mesh generation:
TiberCAD: Automatically generates tetrahedral finite element meshes for given geometry using Gmsh, using Tetgen algorithm.
NanoFEM: Automatically generate tetrahedral finite element meshes for given geometry in Salome, using Netgen algorithm. Generated meshes seem to be obviously better and more suitable then the one which come from Gmsh. The parts in which the mesh should be finer are automatically recognized.

Defining boundary conditions and materials:
TiberCAD: Current version of Gmsh does not allow specifying names of any groups of mesh. Boundary conditions and materials of groups are specified separately in input file of the simulated task.

NanoFEM: Boundary conditions and materials of groups can be specified in the mesh, by correctly naming group of mesh: e.g. *gate[dirichlet=1]*, *bottomOxide[Si]* in Salome user iterface. We plan later to provide an XML file for each task that would collect boundary conditions and more complex parameters like gate functions.

Material database:
TiberCAD: material database consists of plain text files (e.g. “si.txt”) of each material in one file folder.
NanoFEM: material database contained in a standard XML file with XML schema (XSD) allows powerful editing and storing of all characteristics for all materials in one XML file.

Physical models:
TiberCAD: As the result of work of several people dedicated on the project for about 3 years, TiberCAD supports many physical models and tasks that can be solved. It supports 1D, 2D and 3D tasks.

NanoFEM: One developer for about 1 year results in very few possibilities – linear and non-linear solver Poisson (non-linear is furthermore unfinished). However, architecture allows rapid extension, by providing equations in variational form, without necessity to write finite element forms by hand (because they are generated automatically). It supports only 3D tasks.

Control flow of the computing modules:
TiberCAD: The computation flow of various modules is very simple – it is not possible to use flow of modules with anything more complex then a simple sequence and a definition of repeatable sequence of computation, but only with one condition, which is reaching certain difference in precision of computation – i.e. reaching convergence of the task. Therefore any advanced control of module flow is currently not present. This flow is defined in the input file in text format, where most other information related to the simulated task is also defined.

NanoFEM: Salome contains excellent and much more advanced possibility to control flow and connection of simulation modules using supervision or python scripting functionality, or both combined. This way conditions, loops and switches can be defined easily in the graphical schema, or using Python scripting. This graphical representation allows interactive modeling, clear visualization and understanding and debugging of the simulation flow. Modules can be furthermore run on different machines and can run in parallel.

Exchanging of data fields of the running modules:
TiberCAD: Each TiberCAD module contains method, which is able to retrieve previous computed solution. Data to the next running module is retrieved by passing instance of previous module to the current. This module then runs a virtual method to get the whole solution or its parts with passing strings which specifies what should be retrieved, e.g. potential. The passed string is then converted to identification number and further processed in the previous module.

NanoFEM: There is not yet implemented any advanced mechanism of exchanging fields from different modules, however, Salome control flow allows much more powerful mechanism: we can automatically export routines for obtaining solution from the module: e.g. *getPotential*, and we can link this method to the input of next module in supervision editor. The input and output parameters in this case can be Salome field (i.e. FIELD type). When adding module node to Salome, we have list of all available routines in user interface and we are sure that the method exists and is supported. We therefore do not have to use any ID's or strings to pass to some general *getSolution* method, like in the case of TiberCAD. Neither we have to use virtual methods. Instance of module do not have to be passed to the previous module, because we could easily exchange Salome FIELD structure. All these functionality is based on using CORBA based automatic wrappers of Salome. These features are available and are already implemented in Salome, although NanoFEM does not use them right now, because control flow of modules were designed but not tested.

Finite element library:
TiberCAD: Is based on libMesh library, a framework for numerical simulation of partial differential equations. Supports adaptive mesh refinement.

NanoFEM: Is based on Fenics (with C++ interface Dolfin); features include automatic assembly from variational formulation, automatic generation of finite element forms, adaptive mesh refinement and a modularised and flexible programming interface. Defining C++ solution for a new or modified partial differential equation seem to be considerable easier and faster then with libMesh.

Performance:
TiberCAD: To compute linear solver on finite element mesh of a rather complex MOSFET(FinFET) structure, with about 60.000 elements and 15.000 nodes, takes 7.6 seconds. Tested on dual core machine with 3Ghz. During the task, linear matrix system was more complex, because of preparation for eventual computation of drift and diffusion.

NanoFEM: This task takes 5.0 seconds. Tested on a laptop with Celeron M machine, 1.6Ghz. However, the linear matrix was less complex without preparation for drift and diffusion. During the exchange visit, we were not able to setup a more precise and fair comparison, so we could not make a clear conclusion about performance of TiberCAD and NanoFEM Platform.

Post-processing:
TiberCAD: Post-processing based on several tools, like Gmsh and Paraview. Export to Gmsh is not currently implemented. Paraview offers fast, OpenGL accelerated visualization on Windows.

NanoFEM: Post-processing is available in Salome, but currently without hardware acceleration because current version of VmWare 6.0, does not support this feature. However, next versions of VmWare should support hardware acceleration.

Installation (in Windows):
TiberCAD: Installation of TiberCAD core in Windows is very easy, by an installer (35MB), which is very fast and easy to use. It makes association to TiberCAD files, execution of these files automatically runs the TiberCAD executable. However, users must install separately all pre/post processors like Gmsh and Paraview.

NanoFEM: To Install NanoFEM platform, one must first install VmWare and then unpack the provided VMWare image (large; about 2GB 7-zip archive) and open it in VmWare. Nothing more have to be installed, because everything is included in the VmWare image. Furthermore, only one application is needed for all post-and pre processing, and that is Salome Platform, which is of course contained in the image.

Web based interface:
TiberCAD team demonstrated a web application “ICode” based on Google Web Toolkit. Thanks to GWT, an application based on Ajax can be developed using standard Java tools such as Eclipse. The Ajax code is generated by compiling Java code for different web browsers. This allows rapid Ajax development, which will run on different browsers. For Ajax development, it is very powerful and useful technology. An interface similar to the ICode could one day become an interface (at least alternative) for running and coupling TiberCAD simulations.

2. INRIA Rennes - Bretagne Atlantique, team BUNRAKU - with prof. Dumont, Campus universitaire de Beaulieu - 35042 Rennes Cedex, France, 13.5 – 16.5. 2008

During this exchange visit, I presented NanoFEM platform with explanation of motivation for computer graphics (Finite element method as a general method for numerical simulation, FEM for computer graphics – i.e. fluids, smoke, Salome Platform FEM environment, Post-processing, visualization and Automation of FEM), Meshing (for FEM). Detailed overview of NanoFEM software architecture was also presented.

Prof. Dumont provided open source codes developed at IRISA demonstrating solution of obtaining computed mesh from Salome, using file oriented library of Salome platform. Mesh is later used for visualization purposes. NanoFEM platform uses also processing of Salome Mesh, however we use higher level of Salome interface (MEDMEM), which is more powerful and easier to use.

Prof. Dumont also explained in detail and provided scientific papers for a project solving problems of faster visualization of large meshes. Static refinement, LOD, and newly developed methods of mesh multi resolution in INRIA on divided mesh groups using mesh splitting on dual graph, connection of different mesh groups were explained. Problems of out-of-core mesh visualization and performance of mentioned methods were discussed. There were also discussions about possibility of use the finite element method for evaluating external forces on a human body in virtual environments, in particular using finite element method automation using Fenics/Dolfin library.