



HAL
open science

Software coupling and Orchestration Tool to the Modeling of Multi-physic Problem

Lounès Mokhtari, Benoît Delinchant, Jean-Louis Coulomb, Tung Le Duc,
Phuong Pham Quang

► **To cite this version:**

Lounès Mokhtari, Benoît Delinchant, Jean-Louis Coulomb, Tung Le Duc, Phuong Pham Quang. Software coupling and Orchestration Tool to the Modeling of Multi-physic Problem. COMPUMAG 2011, Jul 2011, Sydney, Australia. hal-00610887

HAL Id: hal-00610887

<https://hal.science/hal-00610887>

Submitted on 24 Jan 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Software Coupling and Orchestration Tool to the Modeling of Multi-physic Problems

L. Mokhtari, B. Delinchant, J-L. Coulomb, T. Le duc, and P. Pham Quang

Abstract—We present in this paper Scot, which is a modular software solution for weakly coupling models, methods and orchestration of the simulation. The goal behind Scot is to make easier the phase of modeling and optimize the phase of simulation by tuning the solvers simulation parameters. Specifications and composition of Scot will be described. Scot has been successfully validated by two different applications, the PEEC-MoM coupling to the modeling of an electromagnetic device and the magneto-mechanic coupling to the modeling of deformable nano-switch contact NEMS.

Index Terms—Software component, weak coupling, Tuning solvers, Orchestration, PEEC and MoM method, and magneto-mechanic coupling.

I. INTRODUCTION

THE PROBLEM in multidisciplinary systems modeling is that researchers attempt to model all the different parts of the system that don't appear from their research field, so the models they develop can be of poor quality and may contain several errors of modeling.

In this context, it appears more interesting to use specialized tools to carry out the different models and then link them to rebuild the overall representation of the complex system [1].

Scot (Software Coupling and Orchestration Tool) is a software tool with a modular architecture developed for the purpose of coupling weakly models, methods and specialized tools. It includes some interfaces and makes easy to develop others as required. Models written in any language and embedded in Java, can be easily linked to Scot.

Scot has been successfully tested on two applications. The first concerns the modeling of a real electro-magnetic device by coupling the PEEC (Partial Elements Equivalent Circuit method) and MoM (Method of Moments) method with using a Matlab-Scot coupling. The second test example is a magneto-mechanic coupling applied to the modeling of a deformable switch beam NEMS (Nano Electro-Mechanical Systems).

II. SPECIFICATIONS OF SCOT

Coupling strategies are described by two ways in Scot; in the first, the models are implemented directly in Scot and both of the control and the simulation are done on Scot, however, in the second, there are two layers as shown in (Fig. 1). The layer of supervision from where we configure and control the coupling and, the layer of simulation where the

models exchange the unknown variables. When the desired convergence is reached, the layer of supervision calls the layer of simulation to get the results for post processing.

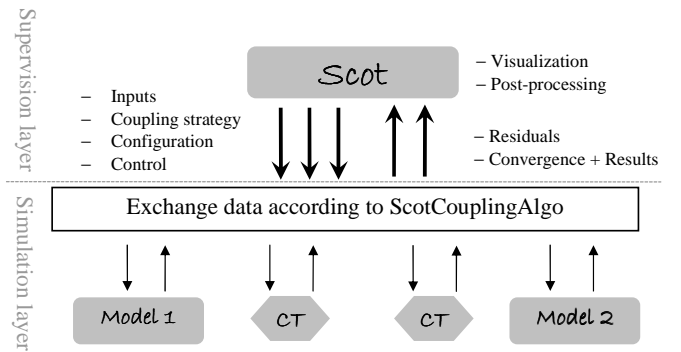


Fig. 1. Scot Coupling scheme of two models using CT (Coupling Terms).

A. Scot elements

1) *Variables*: State variables, are tokens consumed or produced by an ScotComponent or ScotTransformer, characterized by its type (integer, real, boolean...), size, and value. Among Scot variables, there are also parameters of configuration and control.

2) *Components*: ScotSolver is the basic component of Scot, it includes a path to load some type of files, an initialization and solving method, and a name of the recovered variable. ScotSolver is composed by sComp_Matlab and sComp_Java. The first is an interface that allows to couple external Matlab models (Fig. 3(a)), this type of component is used in the application VI-A. The second one calls models written in java language and can use models in ICAR¹ (Fig. 3(b)) components norm as done in the application VI-B.

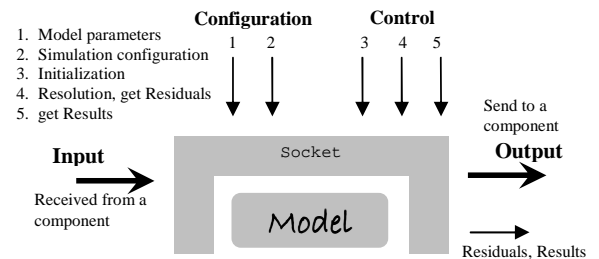


Fig. 2. Representation of Scot Coupling component, its parameters and interactions with Scot manager and other components.

L. Mokhtari, B. Delinchant, J-L. Coulomb, T. Le duc, and P. Pham Quang are with the Grenoble Electrical Engineering Laboratory, BP: 46, 38402 Saint-Martin d'Hères, FRANCE. Corresponding author: L. Mokhtari (e-mail: Lounes.MOKHTARI@G2ELab.grenoble-inp.fr).

¹ICAR : Norm of the software component used in CADES environment [2]

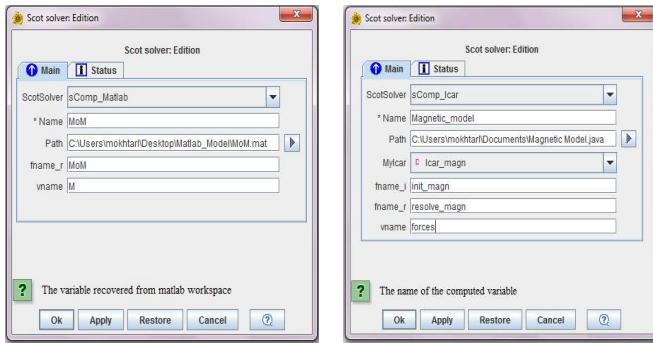


Fig. 3. Graphical user interfaces of Scot Components (a) GUI of Scot Component calling Matlab model, and (b) GUI of Scot Component calling Java class.

3) *Composite component*: The `sComp_Composite` (Scot composite component) is a class of `ScotSolver`, it is specified in the goal to build easily complex coupling algorithms. It allows us to couple `ScotSolvers` sequentially with or without feedback (Fig. 4). With using `sComp_Composite`, some parts of a complex algorithm are encapsulated and seen as a simple `ScotSolver` from the `ScotCouplingTool`.

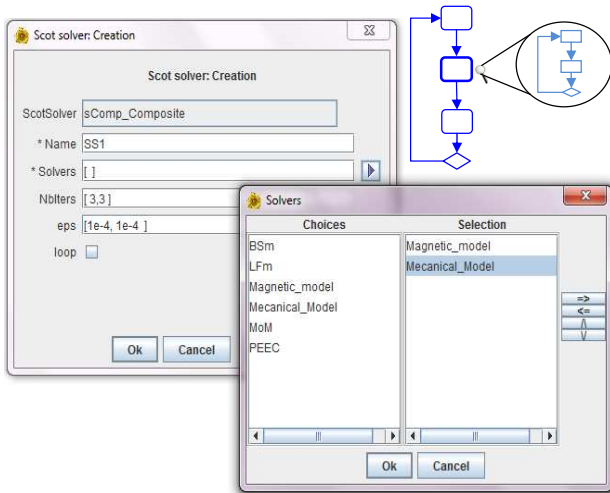


Fig. 4. Graphical user interface of Scot composite component.

4) *Transformers*: Since the models are not necessarily designed to be coupled thus state variables produced by one model can not be consumed by another. In this case, it is necessary to use coupling terms "ScotTransformer". These ScotTransformers may also be transition matrices to reduce the order of the state variables or, software that performs interpolation between different scale meshes such as OASIS [3], MpCci [4].

5) *Coupler*: Represents the manager of the simulation. At this level, we select the coupled `ScotSolvers` and `ScotTransformers`, and specify the orchestration strategy that these selected components will exchange variables. The parameters and functions of orchestration are given by the `ScotCouplingTool`, these parameters define the coupling strategy.

III. UML DIAGRAMS OF THE SCOT SOFTWARE

To facilitate the understanding of the Scot Composition and architecture, we have given in (Fig. 5) and (Fig. 6) the simplified UML diagrams of Scot.

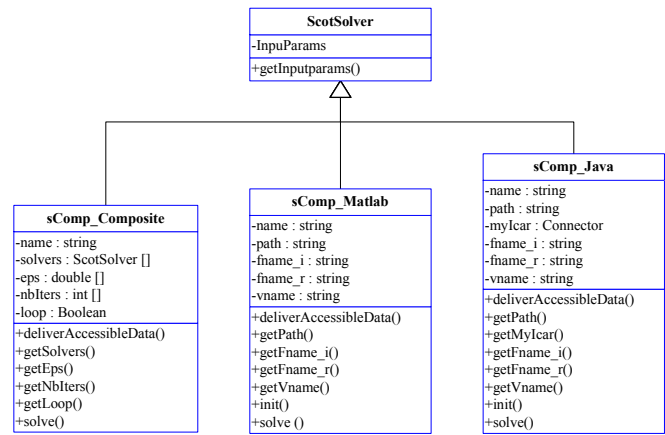


Fig. 5. Simplified UML diagram of the ScotSolver class and its subclasses.

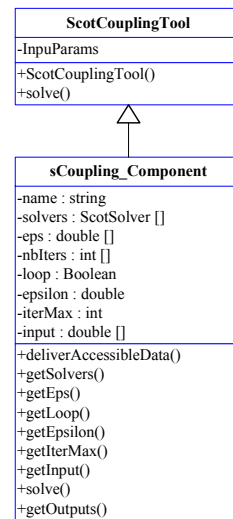


Fig. 6. Simplified UML diagram of the ScotCouplingTool.

IV. COUPLING OF SCOT AND MATLAB

A Socket based communication is developed to couple and adapt the Matlab models to the ScotComponent (Fig. 7), it contains 2 java classes and a ".dll" library and implements the following methods:

- 1) **Initialization** of the Matlab models (build the models according to the model parameters).
- 2) **resolution** of the models and getting the residuals to test the convergence of the coupling.
- 3) **getResults** to recover the final results when the convergence criteria is achieved.

These three methods are respectively called by the `sComp_Matlab` methods : `init()`, `solve()`, and `getOutputs()`. The tight connection between Scot and Matlab is performed by Matlab Engine.

V. COUPLING STEPS DESCRIPTION WITH USING SCOT

Scot has a suitable graphical user interface that allows us to easily build coupling applications by following the steps below:

a) *Step one:* Build the Scot components and Scot coupling terms as explained briefly in section II-A2. These components appear in the work bar in the left side of the main page of Scot.

b) *Step two:* Select a ScotCouplingTool, this one provides as an interface where we can find the ScotSolvers built before, we choose the components to be coupled and give the orchestration parameters.

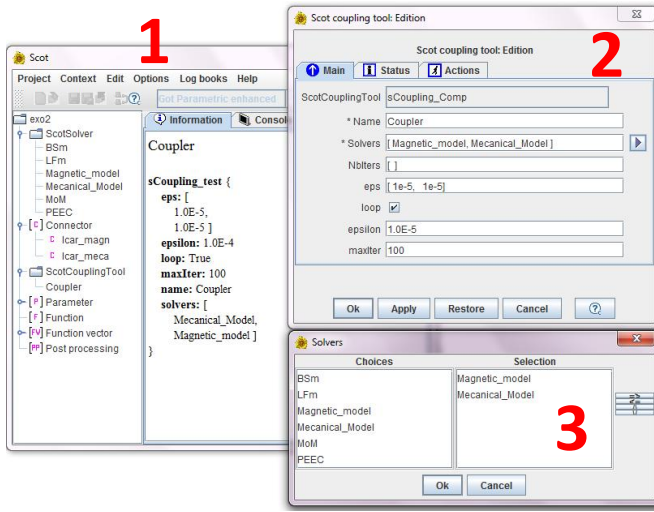


Fig. 7. Steps description of coupling with using Scot.

VI. APPLICATION

A. First Application : Micro-inductance

We have chosen as a first application the modeling of a micro-inductance (Fig. 8) by coupling PEEC and MoM method to respectively model the electric inductance and magnetic materials in the device.

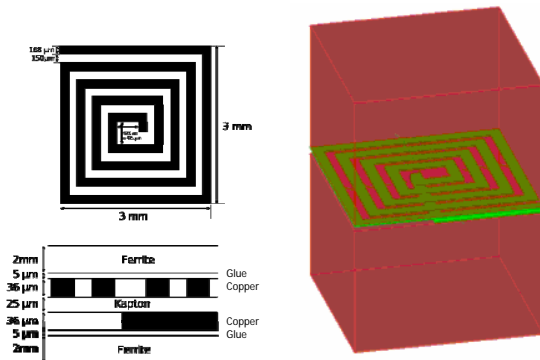


Fig. 8. Micro-inductance, composed by two magnetic regions ($\mu_r = 1000$) and an electric inductance powered by a source voltage ($V = 1$ volt, $f = 1$ kHz).

The numerical model of the micro-inductance is given by the system of equations below (1).

$$\begin{cases} [MM] \cdot [\mathbf{M}] + [BS_m] \cdot [\mathbf{I}] = [0] \\ [Z_m] \cdot [\mathbf{I}] + [LF_m] \cdot [\mathbf{M}] = [V] \end{cases} \quad (1)$$

where,

$[MM]$	Magnetostatic matrix, obtained with LOCAPI [6]
$[BS_m]$	Impact of the electric on magnetic, coupling term,
$[Z_m]$	Impedance matrix, obtained with INCA3D [7]
$[LF_m]$	Impact of the magnetic on electric, coupling term.

The resolution of the system (1) gives us the vector of material magnetizations $[\mathbf{M}]$ and the electric current in the inductance $[\mathbf{I}]$, according to the sources vector $[V]$.

The model of the micro-inductance has been validated with the finite element approach in [5]. For the same results of magnetizations and electric current in the inductance, we have compared the time resolution of strong coupling to the tuned weak coupling proposed in [1] regarding the number of unknowns (Fig. 9).

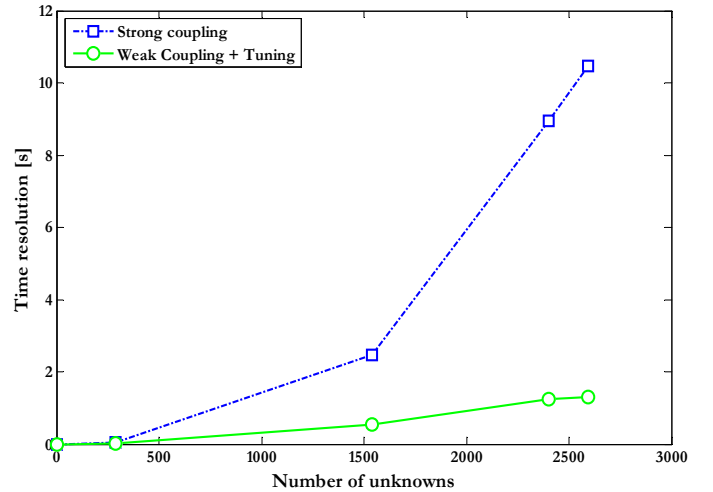


Fig. 9. Curves of the time resolution regarding the number of unknowns.

The curves show that the simulation of the coupling system has found a good interest in orchestration, it has enhanced and sped up the convergence of the coupling.

B. Second Application : Magneto-Mechanic NEMS

The second test case is a model of Magneto-Mechanic nano-switch, composed by a permanent magnet, an electromagnet and a deformable beam (Fig. 10). This model has been build and detailed in [8]. A weak coupling iterative algorithm is used to couple a numerical mechanic model with an analytical magnetic one. The computed variables are the contact forces produced by the electromagnet on the permanent magnet, the beam deformation vector caused by the displacement of the permanent magnet and the contact length with the fixed substrate.

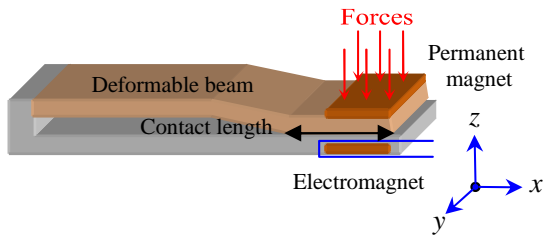


Fig. 10. Deformable Magneto-Mechanic nano-switch NEMS.

To orchestrate the simulation of the coupled model, the available parameters are the accuracy and the number of iteration of the mechanic model, and the accuracy of the integral in the magnetic one since it is solved numerically (Fig. 11).

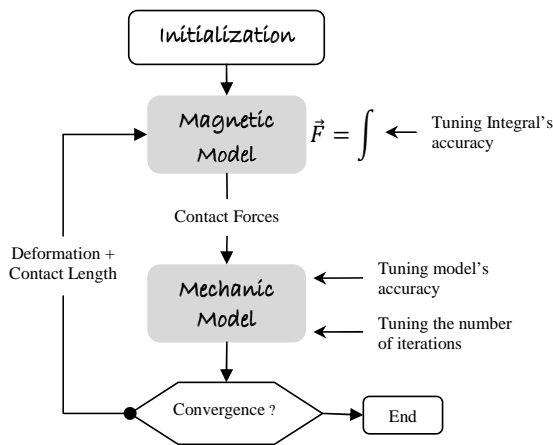


Fig. 11. Iterative coupling model and tuned parameters of the Magneto-Mechanic NEMS and its tunable parameters.

1) *Coupling strategy*: In coupling algorithms, the accuracies of the coupled solvers are unchanged during the simulation. We have proposed a progressive accuracy function based on the coupling iteration number as shown in figure 12,

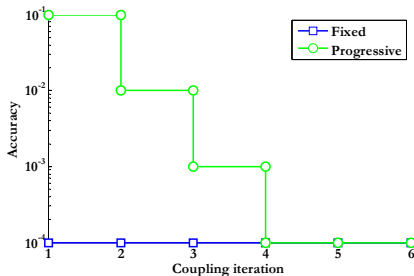


Fig. 12. Progressive and fixed accuracy.

Posteriori parameters analysis can be performed to determine the optimal parameters of the simulation and then automatically tune the coupled model.

A comparison has been made between the fixed and progressive accuracy approach. We have plotted in (Fig. 13) the force and deformation residuals, and given the relative errors and the simulation cpu time of the two approaches.

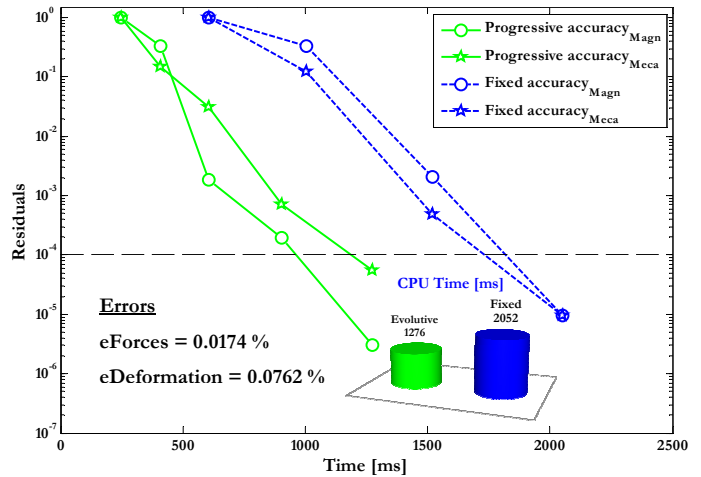


Fig. 13. Residuals of the mechanic and magnetic model, relative errors, and simulation cpu time of the fixed and progressive accuracy approach.

We can see that the progressive accuracy approach is enough faster than the fixed accuracy one, it has converged with a thin relative error and minimized the simulation cpu time by 38 %.

VII. CONCLUSION

We have presented Scot which is a solvers coupling and orchestration tool. The solvers are specified under a norm of Scot Components and have provided the simulation parameters to the manager. These parameters allow us to use the coupling strategies proposed in [1]-[4], develop and test new efficient algorithms to optimize the multi-physic systems simulation.

The concept of layers introduced in Scot minimizes a lot exchanging data with the manager which vacates space memory and requests much less the processor. The architecture based software components used in Scot promotes the reusing of models libraries and so facilitates the modeling of multi-physic systems.

In the applications, we have used a static ScotComponent, our future research concerns the specification of dynamic ScotComponent to the co-simulation of temporal models with variable subcycling step time to accelerate and optimize this type of coupling simulation.

REFERENCES

- [1] L. Mokhtari, B. Delinchant, T. Chevalier, and J-L. Coulomb, *Comparing Weak and Strong PEEC-MoM Coupling*, IEEE Transactions on Magnetics, vol. 46, issue 8, pp. 2775-2778, August 2010.
- [2] [Online]. Available: www.cades-solutions.com.
- [3] [Online]. Available: www.cerfacs.fr.
- [4] [Online]. Available: www.mppci.de.
- [5] T. Le Duc, O. Chadebec, J-M. Guichon, G. Meunier, Y. Lembeye, and B. Delinchant *Numerical experimentations on the coupling between PEEC and volume integral method*, COMPUMAG 2009, Florianopolis (Brazil), November 2009.
- [6] [Online]. Available: www.g2elab.grenoble-inp.fr.
- [7] [Online]. Available: www.cedrat.com.
- [8] P. Pham Quang, B. Delinchant, J-L. Coulomb, and B. du Peloux, *Semi-Analytical Magneto-Mechanic Coupling With Contact Analysis for MEMS/NEMS*, IEEE Transactions on Magnetics, vol. 47, No. 5, May 2011.