International Conference on Adaptive Modeling and Simulation
ADMOS 2009
Ph. Bouillard and P.Díez,

COLLABORATIVE INFRASTRUCTURE TO SUPPORT ADAPTIVE MULTIPLE MODEL SIMULATION

F. DELALONDRE[*], C. SMITH[*] AND M.S. SHEPHARD[*]

[*] Scientific Computation Research Center (SCOREC)
Rensselear Polytechnic Institute
110 8th Street CII-7013, 12180, Troy (NY), USA
e-mail: delalf@scorec.rpi.edu, web page: http://www.scorec.rpi.edu

Key words: Adaptivity, Software Infrastructure, Multimodel, Multiscale, Multifidelity

Summary. This paper presents a software infrastructure being developed by Delalondre *et al.*[1] to support the implementation of adaptive multiple model simulations. It describes an abstraction of single and multiple model simulations into the individual operational components with a focus on the relationships and transformations that relate them. Building on that abstraction, consideration is then given to how adaptively controlled multiple model simulations can be constructed using existing simulation components interacting through functional interfaces. Next, a brief discussion of the software environment used to implement the multiple model simulation infrastructure is given. The infrastructure is finally used to adaptively solve two multiscale and one multiple fidelity model simulation applications.

1 INTRODUCTION

The ability to translate recent advances in understanding the interactions of phenomena across the atomic, molecular, microscopic, and macroscopic scales into new products and industries requires a transformation in the methodologies used by scientist and engineers. Considering the thousands of person-years that has gone into the development, verification and validation of single scale simulation software, an approach that allows that software to be effectively combined is the only practical approach for the development and wide spread application of multiple model simulation.

This paper describes a software infrastructure being developed to support the various classes of area experts that must contribute to the development and application of multiple model simulations. A key feature of the infrastructure being developed is the ability to support a full set of adaptive model, model linking and discretization control techniques.

2 INFRASTRUCTURE DESIGN

The process of defining and generalizing multiple model simulations must begin with consideration of a single model simulation. Figure 1 shows a general abstraction of a single physical model simulation into a set of interacting components. A physics simulation is divided into four hierarchic levels of abstraction. The information defined within each level is placed into one of the three groups: the domain, model and fields. Those groups are related to each other (horizontal arrows in Figure 1) to form the specified level of abstraction.

Abstraction of a multi model simulation is realized by connecting two physics simulations via the one to one linkage of their components. The top level of each model consists in a generalized statement of the problem that defines its conceptual representation. Each of the lower levels represents the result of the transformation, represented by the vertical arrows, that changes the form of the information into the one needed for the next step. The operations involved in the transformations can be complex and involve the introduction of various levels of approximation. The effective identification and adaptive control of these approximations are central to an effective application of simulation.

Figure 2 presents the overall structure that supports the construction of multiple model simulation upon the defined abstraction. It consists of two main parts which are the simulation initialization (left side, Figure 2) and the adaptive simulation execution (right side, Figure 2).

Before an actual simulation can be executed the experts must have provided the system a broad enough set of simulation components and information on how those components interact so that the goals of the simulation to be requested can be evaluated. Assuming a sufficient set of components the application expert begins by indicating the simulation goals in terms of a set of performance parameters to be evaluated and level of accuracy desired for the parameters. The mapping of simulation goals onto the simulation components library allows the creation of a subset of components that are combined to create a hierarchy of physics simulations that can be applied to answer the requested performance questions. The definition of this hierarchy is supported by the use of interoperable interfaces between components. An interoperable component is a procedure that interacts through a functional interface to obtain the needed model, domain and/or field information and performs transformation operations on that information as needed at that point in a multiple model simulation.

Given the simulation goals and hierarchy of physics simulations, the adaptive simulation execution step is responsible for applying the hierarchy that satisfies the defined performance question in the most effective manner. The adaptive simulation execution step is monitored by the simulation state controller that interacts with the built hierarchy of physics simulations, the active simulation state, the adaptive controller and, as needed, the simulation state history. The active simulation state is defined as the physics simulations that have been selected to form the active state of the simulation. The adaptive processes that support this selection ensure the satisfaction of the defined simulation goals throughout the execution of the simulation. Adaptive processes control the errors introduced within the used transformations and select the most appropriate models from the defined hierarchy. This control is achieved by the adaptive controller which is responsible for making the active simulation state evolve, whenever it is required, by realizing the adaptive sequence.

3 IMPLEMENTATION

Functional interfaces of the ten components are used to implement the information passing and transformation routines. The programming language C has been chosen for its ability to efficiently interface other languages. An object oriented data structure defines the organization of the ten components as depicted in Figure 1. The Python programming language has been chosen as it allows the storage of references to the software libraries implementing the ten components.

Introduction of the simulation goals in the infrastructure is through a goal input form. It accepts user knowledge of simulation goals and transforms it into that which can drive the feedback mechanism. Implementation of the feedback mechanism is with an inference engine implemented in Prolog. It provides a mechanism to maintain an evolving set of relationships between the various components and transformations in a manner that allows evaluation of the validity of the utilization of components or transformation based on simulation goals.

High-level control of these processes is through the inference engine that supports the ability to control execution of the a posteriori error measurements and correction indications procedures. The inference engine supports the evaluation of simple rules whereas the component functional interfaces supports computationally intensive routines[2,3,4,5,6].

4 APPLICATIONS

The results of the hierarchic multiscale simulation presented in Figure 3 is obtained by the definition of the simulation goals as macroscopic displacement and microscopic fiber orientation. Selected entities for the active simulation state are macrosopic and/or coupled macro/micro physics simulations[1,7].

The results of the concurrent multiscale simulation presented in Figure 4 is obtained by the definition of the simulation goals as macroscopic strain and energy of atoms at nanoscale. Selected entities are macrosopic (linear and non linear elasticity) and/or an atomistic physics simulations[1,5].

The results of the multifidelity modeling simulation presented in Figure 5 is obtained by the definition of the simulation goals as Von Mises stress state greater than the design limit and joint design parameters. Selected entities are reduced dimension continuum analysis (using shell analysis) and/or continuum analysis that supports non-linearities[1].

REFERENCES

[1] F. Delalondre, C. Smith and M.S. Shephard, “Collaborative Software Infrastructure for Adaptive Multiple Model Simulation”, *Submitted to Comp. Meth. in Appl. Mech. Engng., 2009.*

[2] J. Oden and S. Prudhomme, “Goal-oriented Error Estimation and Adaptivity for the Finite Element Method”, *Comp. Meth. in Appl. Mech. Engng., 41 (5-6) 735-756, 2001.*

[3] J.T. Oden, S. Prudhomme and P. Bauman, “On the Extension of Goal-oriented Error estimation and Hierarchical Modeling to Discrete Lattice Models”, *Comp. Meth. in Appl. Mech. Engng., 194 3668-3688, 2009.*

[4] T. Belytschko and S. Xiao, “Coupling Methods for Continuum Model with Molecular Model”, *Int. J. for Mult. Comp. Engng., 1(1) 115-126, 2003.*

[5] M. Nuggehally, M.S. Shephard, C. Picu, J. Fish, “Adaptive Model Selection Procedure for Concurrent Multiscale Problems”, *Int. J. for Mult. Comp. Engng., 5(5) 369-386, 2007.*

[6] O.C. Zienkewicz and J.Z. Zhu, “The Superconvergent Patch Recovery and A Posteriori Error Estimates, part 1: The Recovery Technique”, *Int. J. Num. Meth. In Engng., 33 1331-1364, 1992.*

[7] X.-J. Luo, T. Stylianopoulos, V. Barocas and M.S. Shephard, “Multiscale Computation for Bioartificial Soft Tissues with Complex Geometries”, *Engng. With Comp., 25 (1) 87-95, 2009.*

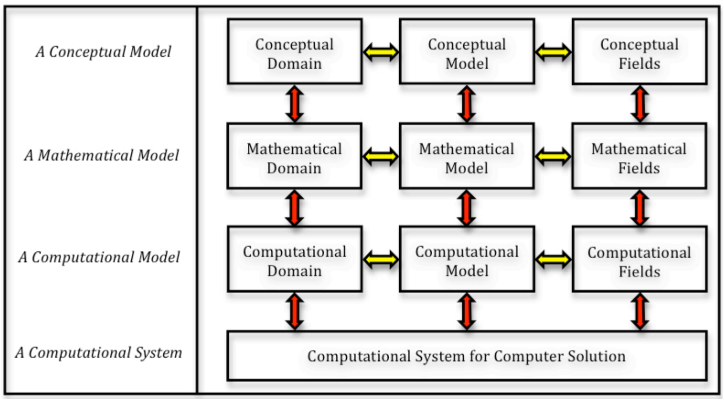

Figure 1: Physics Simulation Abstraction.

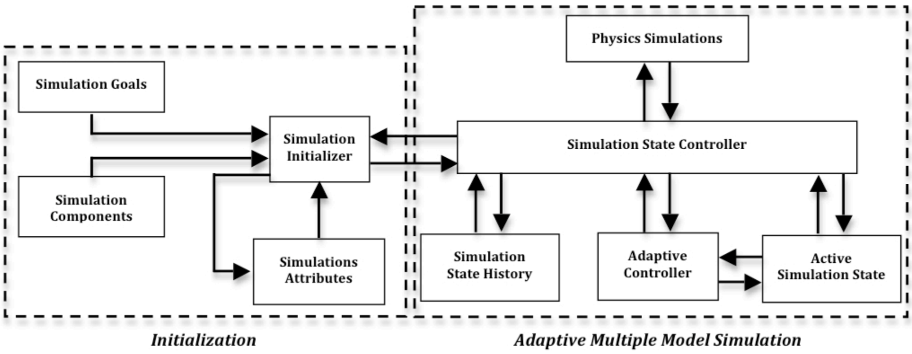

Figure 2: Overall concept.

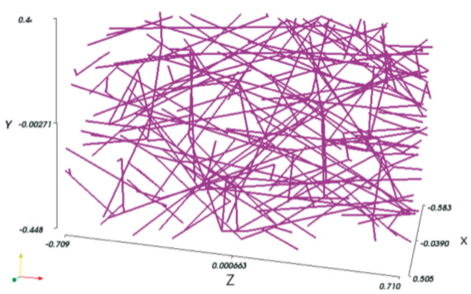

Figure 3: Fiber orientation at load steps 20.

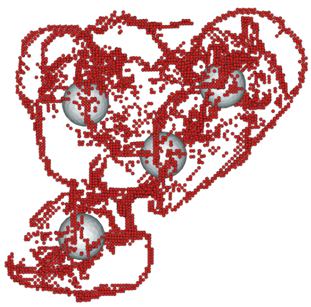

Figure 4: Energetic atoms at load step 28.

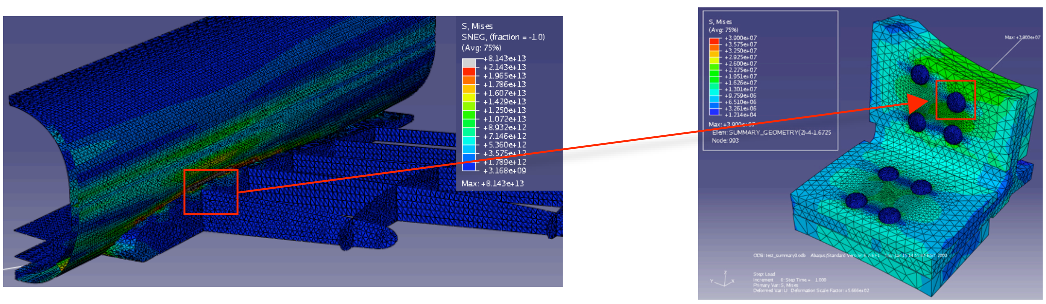

Figure 5: Von Mises stress state in the overall wing structure (left) and a selected region (right).