EEMAS: An Enabling Environment for Multidisciplinary Application Simulations

Lijun Xie^{1,2}, Yao Zheng^{1,2}, Jifa Zhang^{1,2}, Xin Huang^{1,2}, and Zhengge Huang^{1,2}

^{1} College of Computer Science, Zhejiang University, Hangzhou, 310027, P.R. China
^{2} Center for Engineering and Scientific Computation, Zhejiang University, Hangzhou, 310027, P.R. China

Abstract. EEMAS environment is a problem-solving environment for multidisciplinary application simulations. Within the EEMAS, there are four categories of modules involved, namely pre-processing module, computing module, post-processing module, and platform control module. The EEMAS is developed for complex and large-scale simulations to take advantage of powerful parallel and distributed computing technologies. All the modules are coupled through a software bus, which maintains the share memory and makes the modules integrated seamlessly. In the present paper, detailed design principles and applications of the EEMAS are addressed.

1 Introduction

A Problem Solving Environment (PSE) is a computer system that provides all the computational facilities necessary to solve a target class of problems [1]. Typically, it can reduce the difficulty of physical simulation by utilizing user natural languages and application specific terminology, and by automating many lower level computational tasks. We define a kind of PSEs in the following formula [1-3]:

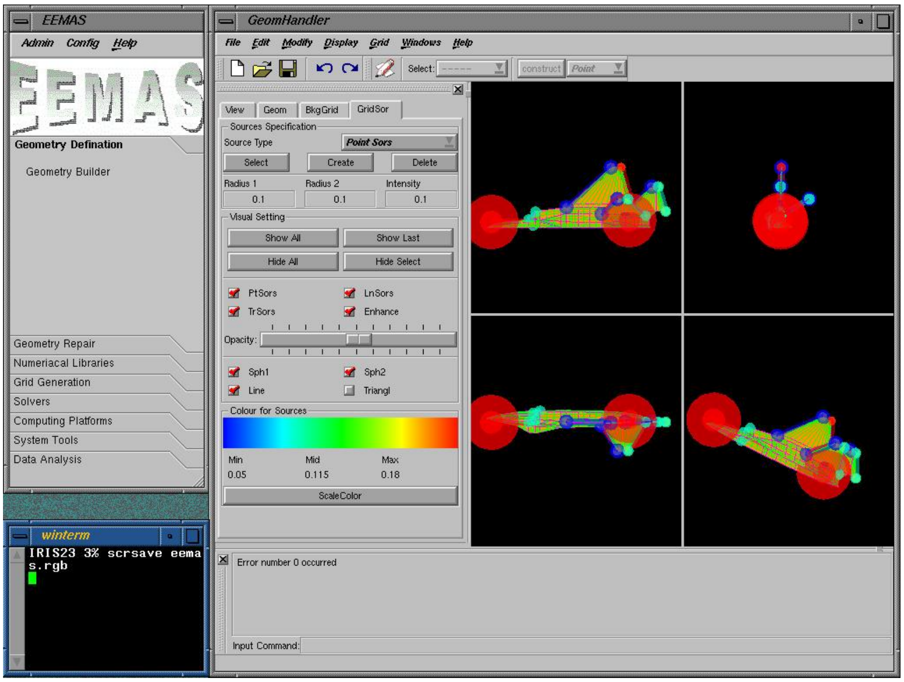
PSE = User interface + Enabling libraries and tools + Problem solvers + Software bus

Commonly, a PSE should have a friendly user interface such as nature language and graphical user interface that can help the user to use the system in a direct and efficient manner. Enabling libraries and tools are the most valuable parts of a PSE. They provide all the necessary assistant functions for a simulation, such as geometric modeling, mesh generation, and scientific visualization. Problem solvers are integrated into the computational module, for various problem fields. Software bus is the method to integrate all the modules to work seamlessly and efficiently.

This paper describes a software architecture and implementation of a problem solving environment EEMAS, which stands for Enabling Environment for Multidisciplinary Application Simulations. It incorporates many modules to support large-scale simulations. Its main functions cover pre- and post- processing, computational platform control and the method to integrate all these modules together. The environment can reduce the time required for problem definition and for post-processing, whilst the unified environment is ideal for multidisciplinary design applications, as the data is handled in one consistent format. It hides many aspects of computational engineering that are not of prime interest or relevance to the engineer, e.g. the setting-up and subsequent execution of an application on a parallel platform. The EEMAS is designed mainly for large-scale simulations, and heavily depend on visual steering, parallel and distributed computation. It is designed for multi-disciplines such as aerospace engineering, civil engineering, energy engineering, geosciences and oceanography.

2 Software Architecture and Features

The EEMAS framework is developed with C++, kernel algorithms are implemented with C and Fortran, the graphical user interface is built with Qt, and visualization capabilities are developed based on the OpenGL library. The system runs on Linux/Unix platforms, while its front end is being ported to Windows at this moment. Fig. 1 is a snapshot of an EEMAS session running on an SGI IRIX machine.



```
IRIS23 3% scrsave eema
s.rgb
```

Source Type: Point Sors

Radius 1: 0.1 Radius 2: 0.1 Intensity: 0.1

Min: 0.05 Mid: 0.115 Max: 0.18

```
Error number 0 occurred
Input Command:
```

Fig. 1. A snapshot of an EEMAS session running on an SGI IRIX machine

2.1 Design Goals

The primary task of the EEMAS is to provide an efficient environment that enables scientists and engineers to create multidisciplinary application simulations, to develop new algorithms, and to couple existing algorithms with powerful enabling tools. The main design principles and goals that guide development in the EEMAS project are as follows.

Abundant Functions. The EEMAS contains generic modules that provide necessary functions needed in mesh-based simulations, such as geometric modeling, CAD repair, mesh generation, domain decomposition, scientific visualization, platform control, and numerical libraries.

Scalability and Seamless Integration. As a Problem Solving Environment (PSE), the main aim of the EEMAS project is not to provide concrete scientific computational functions, but to provide an efficient, flexible and yet consistent framework, that facilitates integration of domain solvers and enabling tools. For this purpose, much attention is paid to the scalability of the system, which is mainly implemented with consistent data format and flexible data transfer interface.

The major portion of data is stored in share memory to reduce the consumption of memory and to improve the speed of data transfer. Three data transfer schemes are provided, that is, through pipes, sockets and temp files, respectively. For a module with its source code, users or developers are able to integrate it into the system by means of data transferring through pipes or sockets, of which the efficiency is quite high. If the source code is not available or the users do not want to spend much time on the integration, temporary files can be used for data transferring directly.

Visual Steering and GUI. The EEMAS follows the philosophy of visual steering. Its graphical user interface guides users to utilize particular components without in-depth expertise on those components, and to control the computing in a straightforward way. This allows scientists to use visualization tools while focusing on computational algorithms, and allows programmers to create visualizations tools without implementing simulation modules either. Visualization and numerical feedback are used throughout the system.

Parallel and Distributed Computing. The EEMAS is designed mainly for large-scale simulations. Therefore, majority of its modules utilize parallel and distributed computing, and can run on remote machines. Most of the modules, from mesh generation to problem solving, and to visualization, can run in the parallel mode. A module for parallel environment setup and control is also developed. Two parallel schemes are utilized. One is task parallelism that distributes different parts of a simulation to different processors. The other is data parallelism that is more widely used within a computationally intensive module.

2.2 Architecture and Functions

Fig. 2 describes the EEMAS architecture from users' perspective. There are four categories of modules involved, named as pre-processing module, computing module, post-processing module and platform control module. The first three are common phases of a simulation whilst the last one serves for the entire process of simulation. All the modules are coupled through a software bus, which maintains the share memory and makes the modules cooperate seamlessly.

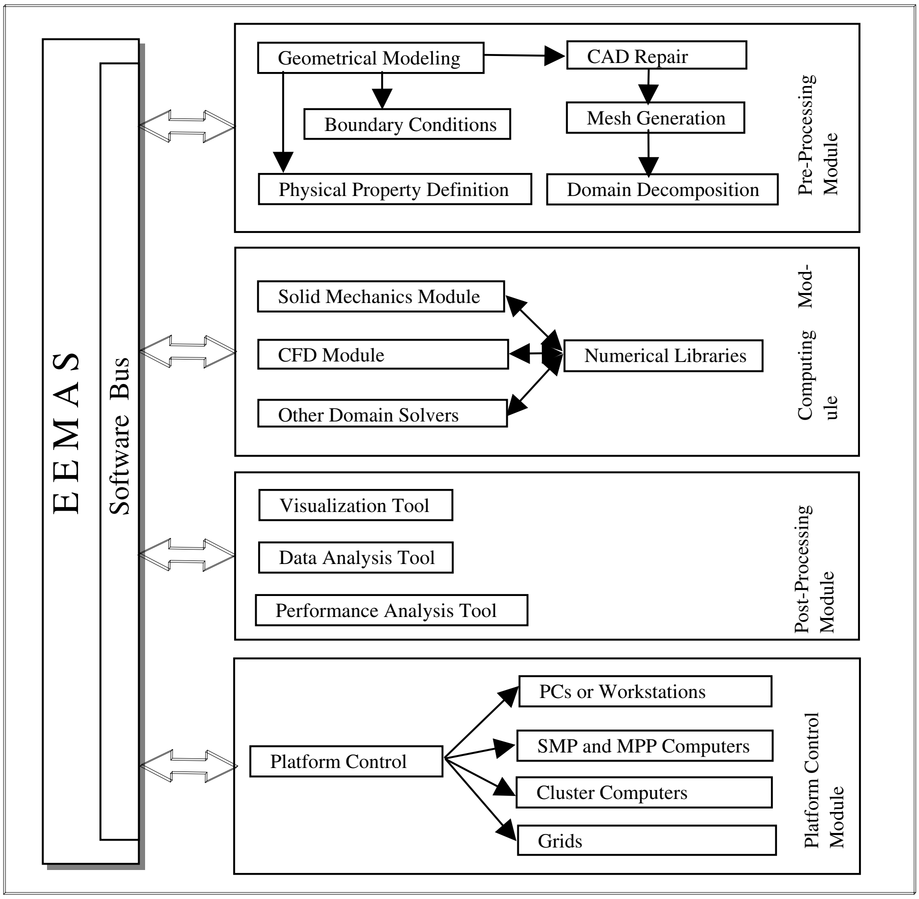


Fig. 2. The EEMAS architecture from users' perspective

2.2.1 Pre-processing Module

Pre-processing module deals with the problem definition and preparation for computing. This is the most time-consuming part in simulations, and many researchers are focusing on automating this work. In the EEMAS, the pre-processing module consists of a basic geometrical handling tool, a powerful mesh generators, and tools for general CAD format conversion, boundary condition definition and physical properties definition.

The geometrical handling tool in the EEMAS stores geometrical data by means of boundary representation. It is not as powerful as commercial CAD software, but it is adequate to process the common modeling work, especially with certain functionalities oriented to mesh generation. For complicated geometries, the user can directly model them with other CAD software, and then import them into the EEMAS.

Mesh generation is a crucial step in simulation that impacts both the calculating time and the accuracy. Research on mesh generation technologies is challenging, while its importance is evident. The EEMAS provides powerful mesh generators with the ability to generate 2D planar meshes, surface meshes of triangles, and volume meshes of tetrahedrons. This module is capable to generate high quality meshes by means of visual steering. Users can specify the mesh spacing in terms of a background mesh and mesh sources.

Mesh generation has a stricter requirement of validation to geometrical models than common CAD software does. It is often the case that the format of geometries inside CAD packages does not meet the requirement for mesh generation. Holes, gaps, overlaps, and intersection of surfaces must be dealt with before mesh generators start. However, issues on geometry validity are not well defined, and it is still a large research field today. In the EEMAS, a geometrical validation and repair module is integrated, so as to find out many common cases of invalid models.

2.2.2 Computing Module

As mentioned above, the goal of designing the EEMAS environment is to support multidisciplinary application simulations. In general, scientists and engineers could integrate various domain solvers into the EEMAS to process particular simulations. To integrate with the EEMAS, users should adapt their solvers with the EEMAS data structure, or write an interface to convert formats. Data transfer involved can utilize pipes, sockets or temporary files. At present, an FEM solver for solid and structure analyses has been integrated.

2.2.3 Post-processing Module

Resulting data of complex and large-scale simulations are often difficult or even impossible to be understood without the assistance of visualization. Two powerful visualization packages named OpenDX [4] and ParaView [5] have been integrated into the EEMAS.

In order to support visualization for large datasets, such as gigabyte datasets visualization, parallel and distributed visualization technologies are employed. There are two modes of distributed visualization. One is the task parallel mode, in which different part of visualization pipes are processed by different processors. The other is data parallel model, in which the data is broken into pieces to be processed by multiple processors. The second method is easier to be implemented, because many visualization algorithms need not much change when running in parallel. The EEMAS supports both distributed and local rendering, and their combination. This provides scalable rendering for large data sets without sacrificing performance when working with smaller data sets.

For a parallel program, programmers and users usually have to tune the parameters and methods repeatedly to get a good performance. Thus a performance analysis tool is necessary. The EEMAS uses a tool named ParaGraph to visualize the behavior and performance of parallel programs on message-passing parallel computers. ParaGraph provides several distinct visual perspectives from which to view processor utilization, communication traffic, and other performance data in an attempt to gain insights [6].

2.2.4 Platform Control Module

The EEMAS provides a platform control module to process the setup of parallel environments, computing source management, file transfer and so on. This hides lot of trivial details of platforms from users, and helps users utilize all kinds of computers from local PCs. The whole system only needs to be initialized once. At present, this module can be used to explore local or remote Unix/Linux systems running on personal computers, workstations, SMP and MPP supercomputers, and clusters. The functions to utilize grid resources are also under development [7].

3 Applications

As an example, a structure analysis of a crank is processed within the EEMAS. After geometrical modeling, the crank is decomposed into 16 domains and delivered to different processors for parallel execution, and at last the data is visualized. We conduct this simulation on an SGI Onyx 3900 supercomputer and a Dawning PC cluster with diverse parameters to explore the performance of the system.

Fig. 3 shows the corresponding geometrical model, discretization of the model and the simulation result. The simulation has been carried out with various numbers of elements and those of CPUs on both the supercomputer and the PC cluster. Fig. 4 shows the efficiency for cases with the numbers of elements and CPUs. From the figure we can see that parallel processing is capable to reduce the computing time greatly. However, more CPUs cannot always guarantee higher performance. When too many CPUs are utilized, domain decomposition, data transfer and other communication operations will take lots of time and reduce the performance.

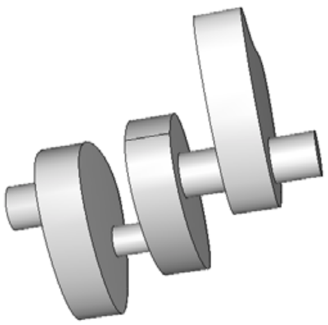
4 Conclusions and Future Work

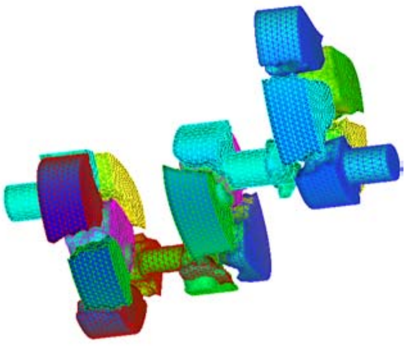
The EEMAS is a problem-solving environment for multidisciplinary application simulations. It is a framework that can easily integrate arbitrary modules, and it contains various enabling tools such as pre-processing, post-processing, platform control and so on. It is especially designed for complex, large-scale simulations to take advantage of powerful parallel and distributed computing technologies.

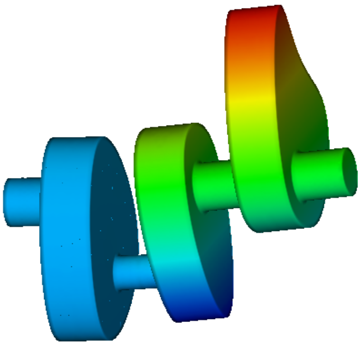
The major work in the future is to extend its applications, by integrating more domain solvers to deal with special simulations. At present, an FEM solver for solid and structure analyses has been integrated, and the inclusion of a fluid dynamics solver is underway. Other modules, such as stereo distributed visualization and platform control tool, are also being improved.

Acknowledgements

The authors wish to thank the National Natural Science Foundation of China for the National Science Fund for Distinguished Young Scholars under grant Number 60225009.


(a) Geometrical model


(b) A mesh and 16 domains concerned


(c) Displacement contour

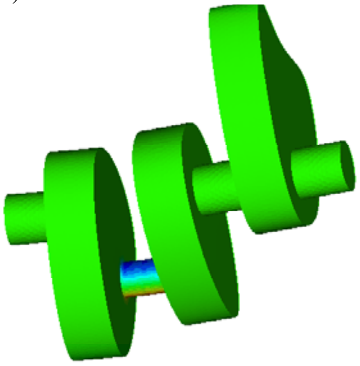
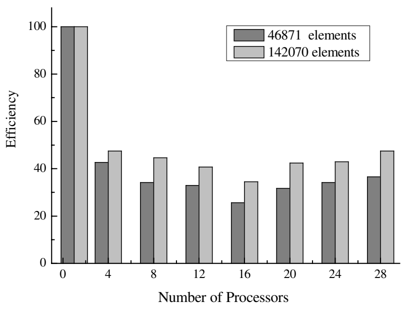
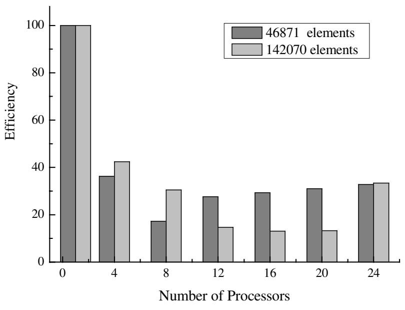

(d) Deformed shape with stress contour

Fig. 3. Simulation of a crank



(a) SGI Onyx3900 supercomputer



(b) A Dawning PC cluster

Fig. 4. Efficiency for cases with various numbers of elements and those of CPUs

References

1. Gallopoulos, E., Houstis, E.N., Rice, J.R.: Future Research Directions in Problem Solving Environments. Technical Report, Department of Computer Sciences, Purdue University (1992) 92-132
2. Rice, J.R., Boisvert, R.F.: From scientific software libraries to problem-solving environments, IEEE Computational Science and Engineering, 3 (3) (1996) 44-53
3. Rice, J.R.: Future Challenges for Scientific Simulation. Chapter 27 in Enabling Technologies for Computational Science: Frameworks, Middleware and Environments, The Kluwer International Series in Engineering and Computer Science, Vol. 548, Houstis, E. N., Rice, J.R., Gallopoulos, E., and Bramley, R., eds., Kluwer Academic Publishers, Boston (2000) 7-19
4. OpenDX User's Guide. URL: http://www.opendx.org/started.html (Current January 1, 2004)
5. Paraview User's Guide. URL: http://www.paraview.org/HTML/Features.html (Current January, 2004)
6. Heath, M.T.: Recent developments and case studies in performance visualization using ParaGraph. In Haring, G. and Kotsis, G., editors, Performance Measurement and Visualization of Parallel Systems, Elsevier Science Publishers, Amsterdam (1993) 175-200
7. Wei, G., Zheng, Y., Zhang, J.: Grid Service-Based Parallel Finite Element Analysis, Proceedings of the Second International Workshop on Grid and Cooperative Computing (GCC2003) (Shanghai, China, 2003), Grid and Cooperative Computing: Second International Workshop, GCC 2003, Part I, Lecture Notes in Computer Science, Vol. 3032, (Li, M., et al, eds.), Springer-Verlag, Heidelberg (2004) 123-130