The Software Architecture of a Problem Solving Environment for Enterprise Computing

Xu Jun Gang, Wang Hong An, and Dai Guo Zhong

Chinese Academy of Sciences, Institute of Software,
Intelligence Engineering Lab., P.O. 8718, 100080 Beijing, China
xujungang@sina.com,wha99@sina.com,
guozhong@admin.iscas.ac.cn

Abstract. Problem Solving Environment (PSE) is a new and diverse area of research, and till now it resists simple universally accepted definitions. In brief, PSE is a complete and integrated computing environment in which new application can be constructed, compiled and executed. This paper describes the concept and software architecture of a PSE for Enterprise computing (E-PSE). The E-PSE is developed to provide transparent access to heterogeneous distributed computing resources in one enterprise, and is to improve research productivity by making it easier to construct, compile and run computer applications so that the enterprise can achieve more advanced economic benefit as soon as possible. The E-PSE includes four modules: Visual Enterprise Modeling System, Visual Application Construction Environment, Application Wrapping & Integrating Tool and Application Runtime Environment. The application of the E-PSE in an oil-refining enterprise is presented. Related work about PSE is also described.

1 Introduction

The current advances in networking protocols (including ATM and fast Ethernet), software development tools, and emerging WWW technologies have enabled the development of a cost-effective, high performance distributed computing environment, network-based computing [1]. But, the available software tools for a high-performance computing environment still require detailed understanding of the underlying architecture and components of application. Writing a parallel and distributed program overwhelms most of the users due to the complexity of communication and synchronization issues [2]. Furthermore, information technologies have almost been applied into every field, but it is impossible to require scientists and engineers in each field to learn computer knowledge and programming. In order that scientists and engineers can devote themselves to research work of their own field, and improve research efficiency and achieve advanced research production, it is necessary to develop an application developing environment for their field. This is the problem which computer experts and experts in the other field face.

This research is supported by major research issue (79931000) of National Natural Science Foundation of China.

P.M.A. Sloot et al. (Eds.): ICCS 2002, LNCS 2329, pp. 862–871, 2002.

As advanced programming language replaced the original machine language owing to shielding the computer hardware, it is believed that there must exist one tool that can shield computer hardware and software to replace current complicated programming work, and can be used to develop application for any field. According to this problem, computer experts have done much research work. Now we have had the answer, which is to develop Problem Solving Environment for each field.

In engineering application field, there also exists the problem that enterprise supervisors and engineers don't know much about information technologies. How to apply information technologies into enterprise better, and how to make use of the advantages that enterprise personnel are very familiar with the enterprise business, which are the problems that computer experts and enterprise engineers need to solve. With the development of global economy, the traditional business mode of enterprise is increasingly confronted with the challenge of market globalization. Day after day people are realizing the importance that information technologies can reengineer traditional enterprise business process. And enterprise must organize and regulate production quickly, achieve the whole optimization of business process, and improve the reactivity and competition capability so that it can stand at an advantageous station in market competition, and achieve more economic benefit. So, to develop PSE for enterprise computing is very necessary.

According to some applications of PSE in engineering field, this paper puts forward the concept of a PSE for Enterprise computing (E-PSE), and describes its features and software architecture. The E-PSE includes four modules: Visual Enterprise Modeling System (VEMS), Visual Application Construction Environment (VACE), Application Wrapping & Integrating Tool (AWIT) and Application Runtime Environment (ARE).

This paper is organized as follows. Section 2 presents the concept, main purpose and features of the E-PSE. Section 3 describes the software architecture of the E-PSE and modules that the E-PSE includes. Section 4 describes the application of E-PSE in an oil-refining enterprise. Section 5 introduces some related work. Section 6 contains conclusions and prospects.

2 The Concept and Features of the E-PSE

According to some applications of PSE in engineering field, this paper brings forth the concept of an E-PSE. An E-PSE is defined as follows, "An E-PSE is one computer-integrated platform for enterprise computing, and it is a complete and integrated Problem Solving Environment in which enterprise application system can be constructed, compiled and executed. It provides various computing tools that can solve problems existed in enterprise, which make supervisors and engineers of an enterprise keep away from the ignorance of computer technologies and devote all themselves to product development, business process management and improvement."

The main purpose of the E-PSE is to provide a model-driven and component-based, application integrating and developing environment. Firstly, through enterprise business process and application modeling, the E-PSE can realize the seamless integration from enterprise business process to enterprise application, and support the integration of the existing enterprise applications and development of new enterprise applications. Secondly, the E-PSE provides a set of wrapping and integrating tools for the existing and legacy applications of an enterprise to take full advantage of the existing resources and protect the previous investment. Thirdly, the E-PSE provides a component-based application development method. One application can be represented as one application flow graph that is composed of some icons of component that can realize certain business logic. Fourthly, the E-PSE provides a set of component repository for certain applications, and the component repository can be extended at any time when it needs. Furthermore, the model-driven and component-based application integrating and developing method provided by the E-PSE can meet the requirements of continuous improvements of enterprise business process. In the E-PSE, business model describes enterprise business process, and application model describes the implementing method of an application. When business process changes, the only thing to do is to change the components or the links between the corresponding components, then the change of the application can be implemented, which diminishes the influences of changes of business process on the relative applications as possible.

The E-PSE has main features as follows: 1) It can make computer resources (including computer hardware, networks and legacy systems) in one enterprise get used sufficiently. 2) It can fit the changes of business object of the enterprise, including the changes of business process and improvement of technical implementation. 3) It can relieve supervisors and engineers from tedious computer technologies and make them concentrate on new management concepts, optimal business practice and process technology, which can improve production efficiency to the maximum extent. 4) It is not necessary to program for application developer to construct new application system, what he needs to do is just to transfer his thoughts into application flow graph in E-PSE, and to improve it in the future continuously.

3 The Software Architecture of the E-PSE

The E-PSE is composed of four modules: Visual Enterprise Modeling System (VEMS), Application Wrapping & Integrating Tool (AWIT), Visual Application Construction Environment (VACE) and Application Runtime Environment (ARE). The VEMS is used to build and modify enterprise model. The AWIT wraps third party application, standard component repository and legacy system into CORBA components, and places them into Component Repository in the VACE. The VACE parses the enterprise model and constructs the corresponding application model and then utilizes the existing software components in the Component Repository to construct new application system, or improve the existing system. Once a new application is completed by the VACE, and then it is delivered to the ARE and is scheduled to run on distributed computer resources. The software architecture of the E-PSE is illustrated in Fig. 1.

3.1 VEMS

The main function of the VEMS is to build and modify enterprise business model. The VEMS consists of two parts: Modeling Tool and Reference Model Repository. We adapt ARIS (Architecture of integrated Information Systems) Toolset as the Modeling Tool, which was published by IDS Scheer AG Corp., Germany. ARIS modeling concept keeps ahead of the world, which was first brought forth by Prof. Scheer in 1992 *(see Scheer, Architecture of integrated Information Systems, 1992)*. It provides an integrated method to represent one enterprise through 5 kinds of views: Function view, Data view, Organization view, Process view and Product/Service view.

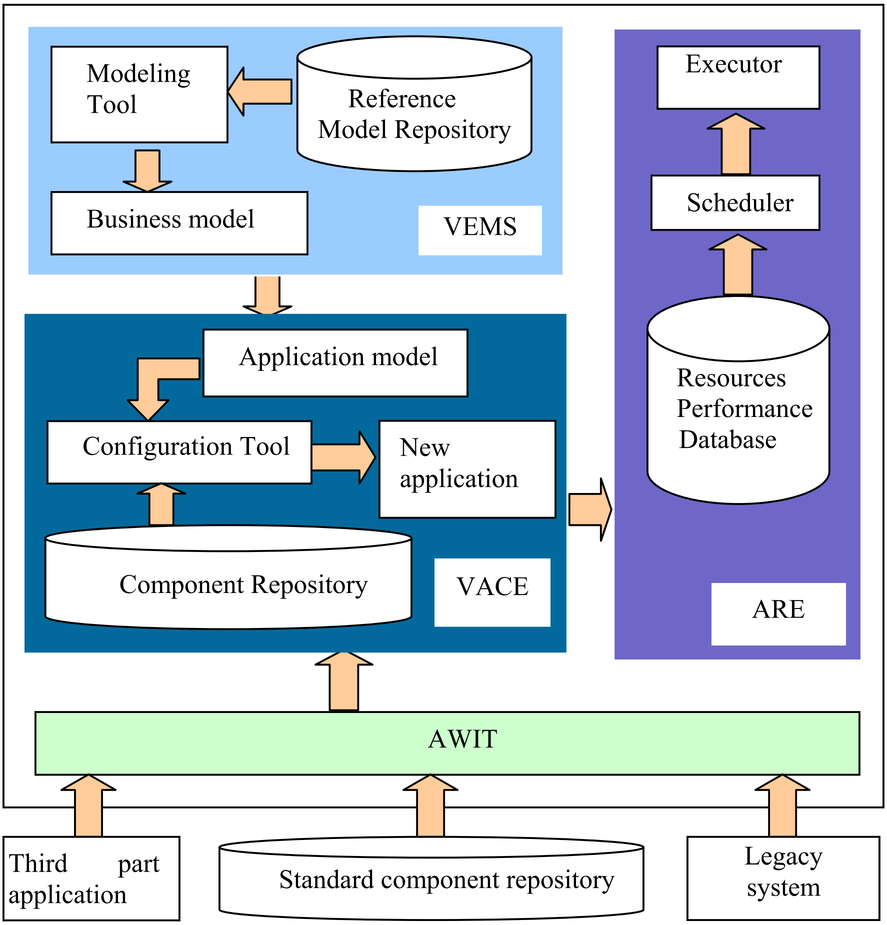


Fig. 1. The software architecture of the E-PSE

When we are building the enterprise business model, we usually look for similar model in the Reference Model Repository first, and then modify it to the required model, it is not necessary to build them from scratch. After customizing one reference model into one enterprise's own model, the enterprise can adjust the customized model according to the changes of market requirements. The Reference Model Repository is based on the analysis of a lot of successful enterprise paradigms and is a kind of knowledge base that collects the experience of many experts in the corresponding industry. The Reference Model Repository describes the specialized requirements of one industry, and so it has comprehensive instructional significance for the enterprise requirements definition and other aspects in that industry. The enterprise feedback information and customer questionnaire make clear that using reference model can decrease workload of defining enterprise requirements dramatically.

3.2 AWIT

The function of the AWIT is to wrap and integrate the existing legacy system, standard component repository and third part application into new application components, and place them into the Component Repository in the VACE so that the VACE can use them to construct new application.

The AWIT adopts CORBA and XML technologies to wrap existing and newly built components. Firstly, the AWIT can be used to wrap a legacy system into a CORBA-compliant component, and then it is placed into the Component Repository in VACE. A complete legacy system may be wrapped as a single component, or it may be divided and wrapped as a series of smaller components on assumption that the internal structure of the system is known [3]. Wrapping legacy system into CORBA object is one kind of wrapping method, which provides convenient interfaces for client to access these objects. The advantage of this method is that the client needn't know actual implementation on server besides the interface. Wrapping can be realized in multi-levels: data, individual module, sub-system and system [4]. After being wrapped into a CORBA object, the legacy system can be reused in form of component in the heterogeneous and distributed computing environment. Secondly, the AWIT can integrate or wrap the third party application, and this needs to negotiate with the third party software provider on related integration matters. After the two parties come into terms, the general adopted method is to convert the API of the third party application into the API of the new application, or change the process of dataflow and workflow in order to exchange information among non-compatible applications. Finally, using XML can simplify the wrapping process. General data format that XML defines allows customer to denote, process and map information among different applications, platforms and interfaces, while these aspects are very critical during the course of processing the legacy system and third party application. The potential value of XML in integrating the existing systems has been paid attention to by many people.

3.3 VACE

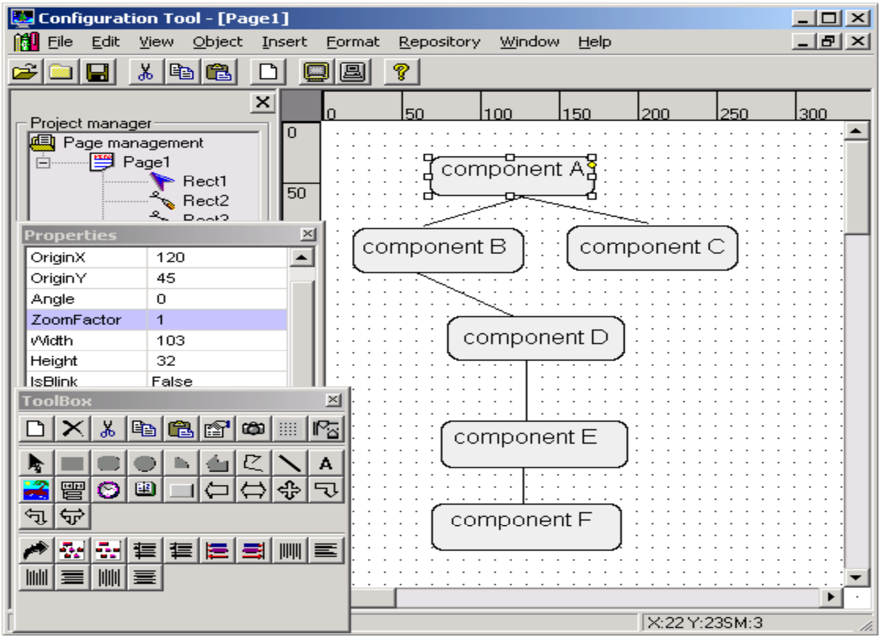
VACE includes two parts: Component Repository and Configuration Tool. The main function of the VACE is to integrate the components that correspond to the business model into a new application by parsing the enterprise business model and utilizing the Configuration Tool. At the same time, the VACE provides the interfaces to real-time database and relative database for applications to realize data accessing.

Component Repository. The Component Repository is a warehouse that stores all kinds of components, mainly including software component and sub-system. The Software component is one basic software object such as COM object, CORBA object, Java Bean and so on, which has its own interface describing file, attributes, methods and events; the sub-system is system resources that consist of some kinds of software components. Users can operate the Component Repository in the same way as operating database, such as retrieving component that has special function, adding and deleting the corresponding component and etc.

Configuration Tool. The Configuration Tool is a visual component composition tool, it makes users create and edit application through composing some kinds of component. In active editor area, a component is represented as a clickable and draggable graphical icon, each such icon includes component name and a set of markers for logical ports. The user can select one new component from the Component Repository and add it into the application sketch, then link it into the application flow graph. In a graph-based programming environment, an application is defined as a directed graph where nodes denote one component and links denote communication and synchronization between nodes. The Configuration tool mainly include key modules as follows:

Project manager
Drawing tool
Control manager
Real-time database manager interface
Script editing environment
Report form defining environment
Variable manager
Page editing interface
Control network interface

Fig. 2 shows the windows of the Configuration Tool. The Project manager implements the operations of creating, deleting and updating of one project. The Drawing tool supports the familiar drawing and rendering methods. The Control manager calls the interface of system registration form, and it can make controls visually display in the page editing interface. If necessary, use the real-time database manager interface to connect with the real-time database manager. The Script editing environment is organized with the format of object, and it provides the method of editing control logic. The Report form defining environment can define the format and content of the real-time report form, which is managed by the Project manager, and executes in the real-time database manager. The Variable manager is an independent module, and it can be installed and published independently, we can use it to execute data gathering and physical variable configuration. The Page editing interface provides one blank template, on which we can build an application flow graph to construct a new application. The Configuration tool can define data source for data gathering and publication through calling the Control network interface.



Properties	
OriginX	120
OriginY	45
Angle	0
ZoomFactor	1
Width	103
Height	32
IsBlink	False

X:22 Y:23SM:3

Fig. 2. The windows of the Configuration Tool

3.4 ARE

The main function of the ARE is to maintain the distributed computable resources of an enterprise, and to schedule the applications constructed by the VACE so that they can run on these resources.

The Computable resources in distributed heterogeneous system should be registered in Resources Performance Database before they are used. The Resources Performance Database maintains information about these resources. The information mainly includes resources attributes and parameters [2][3]. These attributes are grouped into two parts: 1) Static attributes, i.e. Host name, IP address, Architecture type, OS Type, CPU speed, Hard disk and Memory size, Number and type of available network interface, etc.; 2) Dynamic attributes, i.e. recent load measurement, available memory, point-to-point network latency, available network bandwidth, etc.

The Scheduler is the most important part of the ARE. It interprets the application constructed by the VACE and assigns the current best available resources for running the application in order to minimize the total execution time. The scheduling algorithm computes the predicted time of each component and selects the current best available resources within the enterprise networks. The scheduling decision is based on the software and hardware requirements, location and configuration of resources, up-to-date machine/network loads. Once the scheduling decision is completed, the application is delivered to the Executor and runs on the resources under the control of the Executor.

4 The Application of the E-PSE in an Oil-Refining Enterprise

Supported by one National Research Project of China, we utilize the model-driven and component-based method to implement the planning and development of some information systems in an oil-refining enterprise. According to process enterprise automation chain in the ISO specification and the whole framework of the oil-refining enterprise, we present the reference models of the oil-refining enterprise. We first use the Modeling Tool to build the business models of the demonstrated enterprise mentioned above, and then refine the reference business models that fit the oil-refining enterprise from these models. These reference models are helpful for the other oil-refining enterprises to build their own business models. And based on these models, the enterprises can use the Configuration Tool to improve the existing information applications or create new information application, such as the application that can monitor or distribute the process data generating during the production process, and etc. The practicability of these reference models needs proving and improving further in the other oil-refining enterprises in the future.

5 Related Work

PSEs have important significance to the research of each field, so researchers all around the world research PSE in different aspects, such as its infrastructure, reusable components, PSE for special field, or technologies used in PSEs, including distributed collaborative work, visualization, artificial intelligence and so on. The current advances in high-speed networks and WWW technologies have made network computing a cost-effective and high-performance computing environment, which provide more extensive foundation for researching on PSEs.

The current concept of a PSE has its origins in an April 1991 workshop funded by the US National Science Foundation [5][6]. The workshop found that the availability of high performance computing resources, coupled with advances in software tools and infrastructure, made the creation of PSEs for computational science a practical goal, and that these PSEs would greatly improve the productivity of scientists and engineers. This is even truer today with the advent of web-based technologies, such as Java, CORBA and XML for accessing remote resources and sharing information.

In 1994, Gallopoulos ¨ Houstis and Rice presented one detailed definition for PSE as follows, "A PSE is a computer system that provides all the computational facilities needed to solve a target class of problems. These features include advanced solution methods, automatic and semiautomatic selection of solution methods, and ways to easily incorporate novel solution methods. Moreover, PSEs use the language of the target class of problems, so users can run them without specialized knowledge of the underlying computer hardware or software. By exploiting modern technologies such as interactive color graphics, powerful processors, and networks of specialized services, PSEs can track extended problem solving tasks and allow users to review them easily. Overall, they create a framework that is all things to all people: they solve simple or complex problems, support rapid prototyping or detailed analysis, and can be used in introductory education or at the frontiers of science [5]."

Since the 1991 workshop, PSE research has been mainly directed at implementing prototype PSEs and at developing the software infrastructure or "middleware" for constructing PSEs. Initially, many of the prototype PSEs that were developed focused on linear algebra computations [7] and the solution of partial differential equations [8]. More recently prototype PSEs have been developed specifically for science and engineering applications. More generic infrastructure for building PSEs is also under development ranging from fairly simple RPC-based tools for controlling remote execution [9] to more ambitious and sophisticated systems such as Legion [10] and Globus [11] for integrating geographically distributed computing and information resources. The architecture of Virtual Distributed computing Environment (VDCE) developed at Syracuse University [12] is very similar to the software architecture of the E-PSE described in section 3. However, VDCE has not modeling module for enterprise, which is more suitable for computing science than enterprise computing.

As one kind of PSE, E-PSE has many features that common PSE has, and has the similar architecture with other PSEs. However, E-PSE has also some difference from other PSEs, such as PSEs for computational science mentioned above. The difference mainly includes: 1) Common PSEs are usually designed to solve very complex computational problems, and the demands for hardware and scheduling algorithms are very high. But, as to E-PSE, these demands are not very high; 2) E-PSE can support dynamic modeling and continuous improvements of business process of enterprise, which is very important feature of E-PSE. But, many other PSEs didn't have this feature, and it is enough for users to use application modeling function of these PSEs; 3) Common PSEs are mainly used to develop new applications to solve specified problems. However, E-PSE is mainly used to integrate and improve the existing applications, And to develop new application is not major function of E-PSE.

In China, research projects on PSE aren't so many as those in other countries, especially in engineering field. We wish our research could facilitate the development of researching on PSE for engineering field.

6 Conclusions and Prospects

In this paper, we present the concept and the software architecture of an E-PSE. The chief advantage of the E-PSE is that it can relieve the supervisors and engineers of an enterprise from tedious computer application technologies and make them concentrate on new management concepts, optimal business practice and process technologies, which can improve production efficiency to the maximum extent. The E-PSE is composed of four modules: VEMS, AWIT, VACE and ARE.

We have put the E-PSE into practice in an oil-refining enterprise of China, and the effects are very remarkable. But, due to various causes, the theory and method about the E-PSE are just put into practice in enterprise in part, it is expected that the E-PSE would be further popularized and applied in enterprise in the future. And we'll improve these theories and methods further.

If one enterprise wants to keep its invincible position in the situation of more and more intense market competition and globalization, it must adjust its business process according to the changes of the market in time, it is a continuously changing course. The E-PSE adapts a model-driven and component-based software development method, which provides technical support for continuous improvements of business process of an enterprise, and can meet the requirements of high-speed developing information technologies that the enterprise faces nowadays. Consequently, researches on E-PSE are very significant and important for the development of the enterprise, and the application prospects of the E-PSE are also very bright, and it is sure that the E-PSE can bring considerable economic benefits and immeasurable value to the enterprise in the future.

References

1 Panda, D. K., Ni, L. M.: Special Issue on Workstation Clusters and Network Based Computing. Journal of Parallel and Distributed Computing, 40 (1997) 1-3.

2 Hariri, S., Topcuoglu, H., Furmanski, W., Kim, D., Kim, Y., Ra, I., Bing, X., Ye, B., Valente J.: Problem Solving Environments. IEEE Computer Society Press (1997).

3 Walker, D. W.: The Software Architecture of a Distributed Problem Solving Environment. Computer Sciences, 10 December (1999).

4 Otte, R., Patrick, P., Roy, M.: CORBA tutorial, Shixian Li., first edition. Tsinghua Press Beijing (1999) (in Chinese).

5 Gallopoulos, E., Houstis, E. N., Rice, J. R.: Computer as Thinker/Doer: Problem-Solving Environments for Computational Science. IEEE Computational Science and Engineering, Vol. 1 (2) (1994) 11-23.

6 Gallopoulos, E., Houstis, E. N., Rice, J. R.: Workshop on Problem-Solving Environments: Findings and Recommentations. ACM Computing Surveys, Vol. 27(2) (1995) 277-279.

7 Casanova, H., Dongarra, J. J.: NetSolve: A Network-Enabled Server for Solving Computational Science Problems. Int. J. Supercomputing Appl., Vol. 11(3) (1997) 212-223.

8 Houstis, E. N., et al.: Parallel ELLPACK Elliptic PDE Solvers. Intel Supercomputer Users Group Conference, Albuquerque (1995).

9 Arben, P., Sprenger, C., Luthi, H. P., Vogel, S.: SCIDDLE: A Tool for Large-Scale Distributed Computing. Technical Report 213, Institute for Scientific Computing, ETH Zurich (1994).

10 Grimshaw, A. S., Nguyen-Tuong, A., Lewis, M. J., Hyett M.: Campus-Wide Computing: Early Results Using Legion at the University of Virginia. Int. J. Supercomputing Appl., Vol. 11(2) (1997) 129-143.

11 Foster, I., Kesselman, C.: GLOBUS: A Meta-computing Infrastructure Toolkit. Int. J. Supercomputing Appl., Vol. 11(2) (1997) 115-128. See also web site at http://www.globus.org/.

12 Topcuoglu, H., Hariri, S., Furmanski, W., Valente, J., Ra, I., Kim, D., Kim, Y., Bing, X., Ye, B.: The Software Architecture of a Virtual Distributed Computing Environment. Proceedings of the Sixth IEEE International Symposium on High Performance Distributed Computing (HPDC-6), Portland (1997).