GDIXEADP: Service-Oriented Embedded Software Development Platform*

LIU Fa-gui, LIN Kai, CHAI Yang-yang

(School of Computer Science and Engineering, South China University of Technology, Guangzhou 510640, China)

Abstract: In respect of the current variety of embedded development environment, such as multiple processors, peripheral equipments, etc, traditional Embedded Software Development Platform (ESDP) has been unable to meet the demands. A number of development environments, and a variety of development tools had to be adopted to satisfy different needs. That is why to build a unified platform framework for embedded software development becomes an urgent and practical issue. Service Oriented Architecture (SOA) concept provides the possible. Based on the SOA's Service Software Bus (SSB), ESDP framework can integrate a variety of development tools into a unified framework and also possesses a good scalability, and it will be applied to an embedded applications IDE (Integrated Development Environment) for information appliances—the GDIXEAIDE cases.

Key words: embedded software development platform; Service Oriented Architecture (SOA); service software bus

CLC number: TP311 **Document code**: A **Article ID**: 0529-6579 (2008) 02-0037-05

1 Introduction

In order to adapt to the development needs in different embedded environment, due to specific application of different embedded systems and varied hardware (there is a variety of embedded microprocessors, external storage devices, peripherals, embedded components in specialized areas, etc.),[1-2] many application-specific embedded development tools are popular (cross-compiler, debugger, on-line simulator, peripherals communication module)[3]. Traditional embedded IDE has been very difficult to meet the changing demands of the development. The introduction of platform development model to embedded software development is irresistible.

ESDP provides an integrated support environment for the users to develop (including phases of needs analysis, specification, design, coding, testing, product distribution and maintenance, etc.) embedded applications, which includes field-oriented application frameworks, reusable component libraries, reference designs, application examples, the development tool sets, RTOS, related documentations, and other technical facility entities for platform management and configuration, etc.. Its main function is to integrate tools in a unified user interface, provide users with the support of embedded software development[4].

Currently, the popular ESDPs on the market are: Metrowerks's CodeWarrior, WindRiver's Tornado, ISI's pRISM+, Microtec's Spectra, Microsoft's Windows eMbedded Visual C++ and Windows CE Platform Builder, etc.. ESDPs developed by domestic manufacturers mainly are: Beijing CoreTek Systems Incorporated's LambdaTool, and Northeastern University Software Development Center's NEST2000[4-5].

To analyze the current mainstream of ESDP, it is obvious to find that they have the following characteristics: openness, integration, dedication, distributivity, heterogeneity, etc., and can be regarded as a complex distributed application. On one hand, the process of embedded software development becomes increasingly dependent on the support of the development platform; on the other hand, users who are in the use of these development platforms can only use tools which provided by the single vendor, and non-standard mechanisms used by most tools integration constrain the diversity of tools and the the integration the platform.

The key to construct of a new type of ESDP to meet the command of integration, openness, applicability, flexibility, etc., lies in its software architecture. This paper presents a design of service-oriented ESDP. This development platform is based on the SOA's SSB structure, and functions of tools will be integrated into a unified environment in the form of service, which can

* **收稿日期**: 2007-06-26
基金项目: 广东省粤港关键领域重点突破基金资助项目（2005A10207005；信产厅2004-0005）
作者简介: 刘发贵（1963年生），女，教授，博士；E-mail: fgliu@scut.edu.cn

adapt to the development needs in a variety of operating system, and embedded microprocessors, and have rich development, debugging, testing and other functions. Users can construct IDEs in accordance with their needs. Platforms no longer face to a certain specific areas and systems, but an extensible service-oriented system.

The rest part of this paper will introduce the research status, architecture, main technology programs of the service-oriented ESDP, and its application and realization in the information appliance development

2 Related Work

Currently, there are a lot of studies in the architecture design of the ESDP. Using object-oriented, model-driven design methods has been associated with the study. References [6－8] discussed the usage of the object-oriented methodologies and design patterns in the realization and application of the ESDP. As ESDP tends to be more complex, distributed and heterogeneous, object-oriented development technology and model-driven architecture bring a great deal of restraint to the flexibility and scalability of the development platform.

Service-Oriented, soft-bus architecture-based ESDP design has became a hot topic. Reference [9] brings forward a service-oriented software model based on the embedded operating system Elastos. References [4] and [5] established a componental tool bus model, and respectively used CORBA and COM technology as the tool bus to construct ESDP. But these discussions either only concentrate on the application of SOA in the embedded development environment and do not relate to constructing the specific IDE or their programs apply only to the specific hardware and software environment.

Meanwhile, the idea of using plug-in architecture to extend and transform the existing open-source development tools reflects the thinking of SOA and SSB structure at certain degrees, which can also make fully use of the existing open-source tools, save cost and speed up the development process, somehow makes it a good choice for constructing the ESDP. References [10] and [11] adopt plug-in architecture, and extend the open source tools—JEdit and KDevelop respectively to construct a visual embedded software development environment. Reference [12] and [13] discuss the design and implementation of the design and development environment of the embedded software based on the Eclipse in detail. Although absorbing only a certain service-oriented thinking in the design, and all gearing to specific hardware equipment, they still provide a useful insight.

This paper will present a service-oriented ESDP based on the SOA's SSB architecture. It can meet the cmomand of a variety of hardware and software development environments with ability to be extended and customized.

3 SOA and Embedded Software Development

SOA is a new method for software engineering development, and a common method for framework design. SOA is a new architecture based on extending the object-oriented architecture. It supports multiple standards, has a flexible application, and is the development trend of the software engineering. The purpose of constructing a service-oriented ESDP is to provide a multifunctional IDE with higher scalability for embedded application development. Therefore, the platform's software architecture should have some flexibility, in order to facilitate the expansion of new functions, the integration of new tools and the reduction of development cost.

With the development of embedded development model, SOA gradually absorbs many of the software engineering's ideas, and breaks through the traditional embedded software development models. Process-oriented, object-oriented, component-oriented software engineering ideas are widely applied in the area of embedded. So, how to combine the currently ascendant SOA theory with the embedded development environment, and solve the above mentioned problems in the face of the embedded development is a practical and promising research topic.

SOA is a component model. Different functional units (called services) of the application will be linked through well defined interfaces and protocols. Interfaces are defined in a neutral manner, and are independent from the specific platform realization, system and language, commonly in the format of XML. It makes that all services of SOA can interactive in a unified and common way and cause a lower coupling between the services so that the overall architecture of the application will not change along with the individual service's internal structure[14]. Services call each other through service description which is an agreed model used to describe what services are, how to call services and what kinds of data are needed for successful service calls.

With the concept of SOA's SSB, we can conduct a design of a service-oriented, extensible application IDE. SSB is a software bus. Essentially, bus architec-

ture means that all the functional components are connected to an intercommunicating public structural component. Available solutions models of SSB are: basic adaptor, service gateway, web service-compliant broker, etc[15]. This paper adopts the basic adaptor model, to finish the conceptual design. The overall structure of the entire service-oriented ESDP is shown in figure 1.

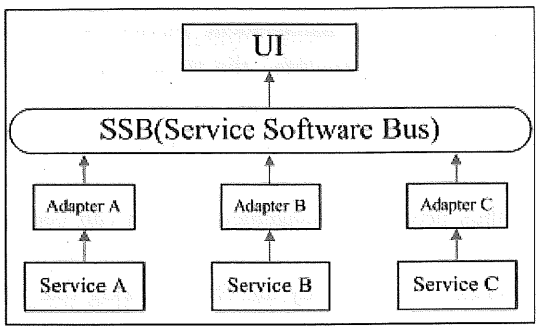

Fig. 1　Overall structure of the service-oriented ESDP

Each development tool of the embedded software platform is wrapped to be service entity, which is connected with the SSB through adaptors. Tools provide development or management functions needed during the development process, and communicate with each other in the way of request-response through the SSB. SSB manages interoperate of the tools. Adaptors are in charge of the information conversion and communication protocol adjustment between service entities and the SSB.

4　Embedded Software IDE Framework Base on SOA

The main problem of designing the embedded software IDE is the design and implement of the software bus. After observing the current popular software development platforms, the author found that Eclipse's framework is more in line with the embedded development needs.

Eclipse is a complete and open basic development platform for tools integration, and released in the way of open source, which provide a series of development tools, and essential functions needed by the IDE for the developer, such as resource management, user interface, team support, help system and so on. Its architecture is shown in figure 2.

Eclipse is a software framework which is designed following the OSGi (Open Service Gateway Initiative) standard, and Eclipse platform is constructed according to the extension point conception. Platform runtime is the core of the entire Eclipse framework, and all the

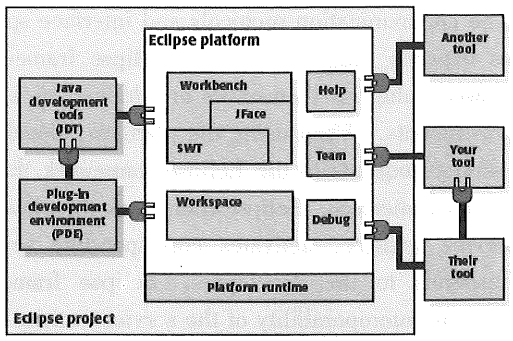

Fig. 2　Eclipse platform framework[16]

other functional parts are implemented in the form of plug-ins to add functions to the platform. The greatest value of Eclipse lies in that it provides a good framework for constructing software development platform. Basing on the plug-in model, we can integrate any essential functions to the Eclipse platform in the form of plug-ins through interface standards defined by Eclipse to extend the function of the platform.

Meanwhile, as an open-source project, Eclipse save the cost of developing an ESDP. The powerful tool integration capability and good cross-platform characteristic makes it raise the level of consistency in different embedded software environments. Developers use different functions provided by the platform in different processors and operating systems, but it can maintain the consistency of the operations and user interfaces[16].

Using of the Eclipse plug-in mechanism and the Eclipse framework, we can facilitate our design of the platform. As shown in figure 3, we use Eclipse framework to implement the SSB. All the functional parts are wrapped and transformed using basic adaptors. They are implemented in the form of Eclipse plug-ins by extending the extension points defined by Eclipse. Plug-ins also make functional extensions with each other through user-defined extension points, to provide a more powerful service. (Adaptors are not paint in the figure)

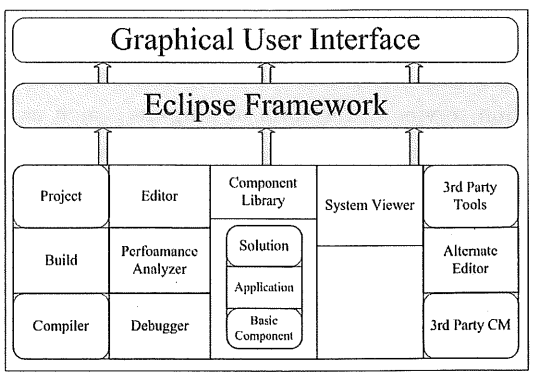

Fig. 3　Design of ESDP

The communication protocols and interface specifications between plug-ins and the Eclipse framework, and between plug-ins themselves are defined by simple XML documents. When using the platform, developers send service requests to the Eclipse framework through graphic user interface. Eclipse framework then loops up the service registry, activates corresponding services and responds to the developers. Eclipse framework manages the interoperability of the service entities. Application modules are well designed, having a very low coupling with each other and the Eclipse framework.

According to the needs of the actual development process, developers can add or delete application modules in the SSB at any time to customize an ESDP which meets the actual command.

5 An Example: GDIXEAIDE

5.1 The component of GDIXEAIDE

Based on the above SOA design, we achieve an information appliance-oriented embedded application IDE—GDIXEAIDE. Figure 4 shows the location of the ESDP in the overall IDE.

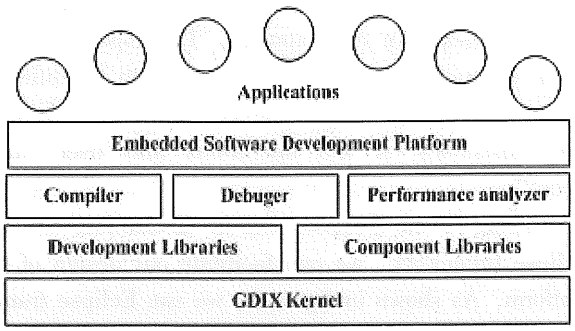

Fig. 4 Structure of GDIX embedded software development environment

At the bottom of the embedded software development environment, there is the operation system GDIX which is compacted by us, and above which, we provide development libraries and component libraries, and integrate tools such as cross compiler, remote debugger, performance analyzer, etc. into it. According to the uniform interfaces, we integrated a number of application service in the form of plug-ins, such as component management system, kernel configure tools, linkage script editor, C/C++ visual development tools, cooperating development and version control, etc.

Practice proved that embedded software development environment which adopts the SSB structure can facilitate the addition or deletion of the service modules, without affecting the other function services. We only need to uniformly implement the interfaces according to the standards of the SSB interfaces and using the adaptor mode, which has a convenient achievement. With the support of Eclipse PDE, service development becomes much easier.

5.2 Addition of the privilege management service module

GDIXEAIDE can satisfy the needs of collaborative development in distributed environment. To achieve this goal, privilege management service is essential. Based on the SSB structure, the author implements the privilege management module in an actual project. Theoretical foundation of the implementation is just that all the application modules are services on the platform, thus privilege management system can communicate with the platform, and manage these services by the interfaces provided by the SSB, making that different users have different privileges of services.

After users login on the platform, certification module of the privilege management proceeds to a legitimacy certification, and then it will query the IDs of all the service the user can access, and compare them with IDs of all the registered services queried from SSB. It sends a request to the SSB, and enables the services that the user has the privilege to access.

Addition and deletion of the privilege management services do not need any modification to the existing service. If we want to add new application service to the development environment, we just need to modify the privilege management system itself, adding the ID of the new service to the setting of the user privilege. From the example, the benefit of SOA is obvious. Coupling between the services and the development environment, and between the services themselves is very low. Users can construct ESDPs suitable for some specific application just as easy as building blocks.

6 Conclusions and Further Study

ESDP using SOA can well meet the needs in different embedded environment. As long as it follows uniform standards of SSB, a number of services and 3rd-part tools can be integrated into the application IDE conveniently.

This paper presents a framework of the ESDP based on the SOA's SSB structure, which have a powerful tool integration ability and good scalability. This paper also described an information appliance-oriented embedded application IDE—GDIXEAIDE example, and shows the advantage of the SOA-based ESDP by the example of the addition of the privilege management system. In practical implementation, we use Eclipse

framework as the SSB standard. Users can customize a fully personalized embedded software platform according to the needs of themselves.

SOA technology is still in development. How to further play to the superiority of SOA architecture, perfect the design of the ESDP, further improve the level and flexibility of tools integration, enhance capabilities of distribution collaborative development, reduce the difficulties of development and so on, are the directions for the future study.

References:

[1] SANG Nan. Embedded systems theory and application development technology [M]. Beijing: Beijing University of Aeronautics and Astronautics Press, 2002.

[2] LEE E A, NEUENDORFFER S, WIRTHLIN M J. Actor-oriented design of embedded hardware and software systems[J]. Journal of Circuits, Systems and Computers, 2002,11:33 - 36.

[3] SUSANNE Graf, JOZEF Hooman. Correct development of embedded systems [C]. EWSA: 2004.

[4] GUO Bing. Research on software architecture of open integrated development platform for embedded application [D]. Chengdu: University of Electronic Science and Technology,2002.

[5] MA Derong. Research and implementation of embedded software development platform TIDE [D]. Chengdu: University of Electronic Science and Technology,2005.

[6] LIU Xiangshuan. Design of embedded application development platform [D]. Qingdao: China Ocean University, 2006.

[7] ZHANG Zhenhua. Research of the design of a real-time embedded systems software development platform [D]. Xi'an: Xi'an University of Technology, 2003.

[8] XIAO Fangxiong, XIONG Guangze, LU Dongxin. Design and realization of an object-oriented embedded software development platform [J]. Computer Applications, 2003, 23:341 - 343.

[9] LU Yi, GU Weinan. On the service-oriented software models based on the elastos operating system [J]. Computer Engineering & Science, 2005, 27(3):83 - 85, 92

[10] YAP N, CHIONG H C, GRUNDY J, et al. Supporting dynamic software tool integration via web service-based components [C]. Software Engineering Conference, 2005, Proceedings Australian, 2005:160 - 169.

[11] FANG Qian. Research of embedded system-oriented development platform [D]. Hangzhou: Zhejiang University, 2005.

[12] ZHOU Xuegong, CHEN Yan, LIANG L. Embedded system integrated design environment based on eclipse platform[C]. Computer Supported Cooperative Work in Design, 2005, 1:113 - 117.

[13] JARVENSIVU J, KOSOLA M, KUUSIPALO, et al. Developing an open source integrated development environment for a mobile device[C]. Software Engineering Advances, International Conference, 2006, 55 - 55.

[14] DENG Zhengjie, CHEN Zhanglong. Designing source-level debug for embedded system platform [C]// Proceedings of The 8th International Conference on Computer Supported Cooperative Work in Design, Guangzhou, 2004, 1:743 - 746.

[15] ROBINSON Rick. Understand enterprise service bus scene and solutions in service-oriented architecture [EB/OL]. [2004 - 07]. http://www-306. ibm. com/

[16] ROBERT Day. Eclipse makes it easier to move between embedded operating systems and target platforms [C]. IET Electronics Systems and Software, 2007:20 - 23.

GDIXEADP：面向服务的嵌入式软件开发平台

刘发贵，林　恺，柴阳阳

（华南理工大学 计算机科学与工程学院，广东 广州 510640）

摘　要： 针对当前多种多样的嵌入式开发环境，如多种处理器、外围设备等的存在，传统的嵌入式软件开发平台已经不能满足嵌入式开发多变的需求，不得不采用多个开发环境、多种开发工具以满足不同的需求，因此，如何构建一个统一的嵌入式软件开发平台架构，是一个迫切而实际的课题。SOA（Service Oriented Architecture）概念提供了一种可能，提出了一个基于 SOA 的服务软总线思想的嵌入式软件开发平台架构，可以将多种开发工具集成到一个统一的框架且具有很好的扩展性，并且将它应用于一个面向信息家电嵌入式应用的集成开发环境 GDIXEAIDE 实例。

关键词： 嵌入式软件开发平台；SOA；服务软总线

中图分类号： TP311