Technology Analysis and Application of Software Reuse to Electronic Information System

Xianyu Li

Beijing Shi QingHe DaLou Zi Jiu, Beijing, 100085, China

email: kking127@sohu.com

Keywords: Software Reuse; Software Reuse Framework; Electronic Information System

Abstract. Based upon the synopsis of software reuse technology, this paper describes the software reuse process and puts forward software reuse framework. Meanwhile, practical analysis is carries out, and platform application mode is deduced that comprises development environment, management environment, operation environment and component library. This will benefit the agile development of electronic information system.

Introduction

Electronic information system is distributed man-machine system that effectively organizes related resources, conducts information processing and actualizes business activities. In order to advance the applicability to former system, the expansibility of new system and continual improvement on business flow, electronic information system need further satisfy the requirements of dynamic function evolvement and swift mission realization, and meantime the practices and development of software reuse technology can provide available agile approaches and evolutive system for it [1][2].

Software reuse is the process during which new software system could be established by means of developed software components, and it clearly benefits the improvement of development productivity, the enhancement of software quality, and the betterment of system maintainability.

The key factors of software reuse mainly comprise the following aspects:

(1) Software component technology is helpful to achieve the basic reusable units;

(2) Software architecture benefits the integration of reusable component;

(3) Domain engineering emphasizes on the production of reusable component in the specified fields;

(4) Software reengineering focuses on the identification and regroupment of reusable component as concerning the prior systems;

(5) Open system technology is devoted to the standardization and mutual operation of reusable component interfaces;

(6) Software process specifies the development process of reusable software component;

(7) Facilities and environment provide the automation support for the technology, methods and process;

And some non-technology factors, such as organization framework, management, psychology and intellectual property rights, have certain and different influence on software reuse.

Software Reuse Process

Software reuse process [3][4] is closely connected with system and software development process, and it generally comprises the activities such as domain analysis, framework design, reuse component creation, reuse component validation and software reuse appraisement. Software reuse activities are carried out along with related software development phases. Software reuse process is shown in Figure 1.

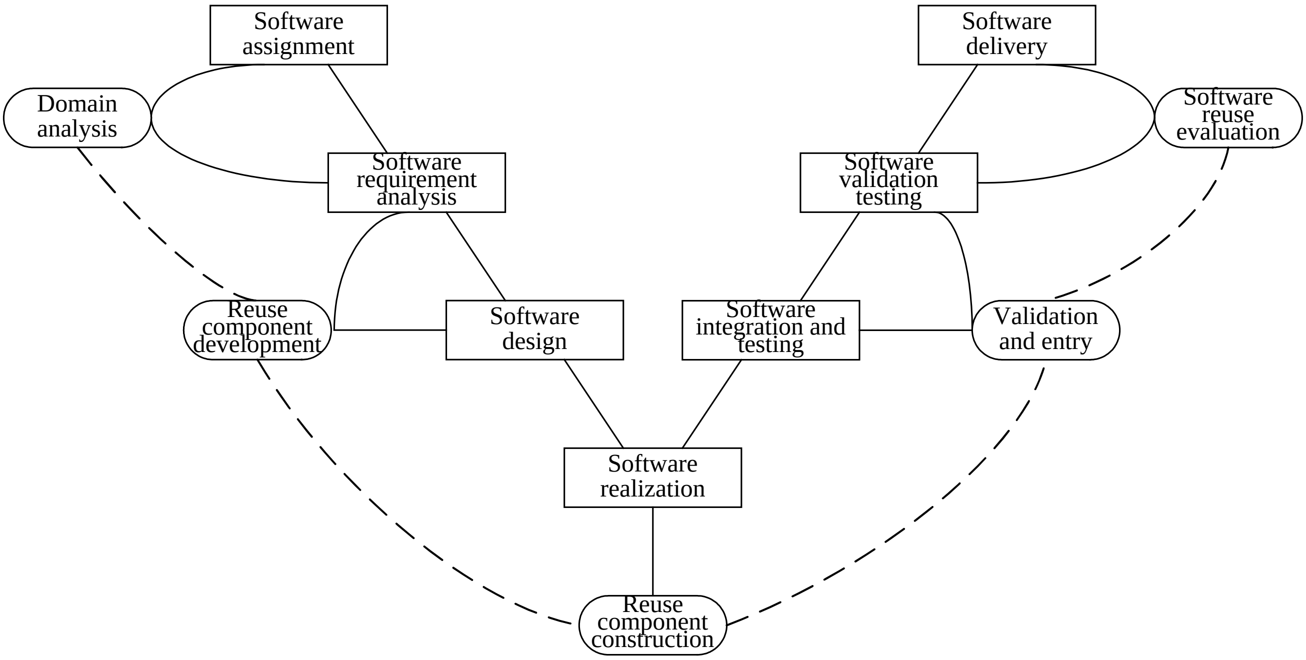
Software assignment
Software delivery
Domain analysis
Software reuse evaluation
Software requirement analysis
Software validation testing
Reuse component development
Software design
Software integration and testing
Validation and entry
Software realization
Reuse component construction

Fig. 1. Software reuse process

(1) Software assignment. Software assignment is stipulated in software development contract that usually comprises demands for software function and performance, development environment, standards, testing, and acceptance criteria. Domain analysis is actualized to sum up common functions and requirements, independent characters, and special requirements within the specified field. It eventually determines the domain model that includes related business functions and data processing, and can be optimized along with the practical status.

(2) Software requirement analysis. The software requirements are obtained by means of requirement analysis and user communication, and comprise the requirements of capability, applicability, environment, quality, training, and configuration item, and eligibility prescription. Framework design is implemented on the basis of domain analysis. System concept structure need to be further translated into logic structure, and collectivity architecture, function and performance of software; answer field model need to be derived, and use case method can be applied into the detailed analysis. Firstly, business requirements are analyzed; secondly, knowledge from domain specialists is refined and deduced into knowledge base; thirdly, inspections are carried out to make sure that reusable software be consistent with domain knowledge base.

(3) Software design. Proper software architecture and configuration item interface design are constituted, and the design elements concerned with system monitoring, diagnosis and revision are available. Component interfaces are generally standardized to have high cohesion and expansibility so that reusable components can be employed and integrated directly. The reusable components of new domain are designed and developed according to the development process, and related algorithm, flow and interfaces are analyzed and made clear in detail.

(4) Software realization. Software coding is implemented according to the reviewed software design, and unit test is completely actualized to ensure the functionality and quality. The software reuse components can be realized as basic and extensible configuration items and basically satisfy practical demands under the typical environment so as to set up the desired software system.

(5) Software integration and testing. Software integration and testing is carried through to improve and validate the accordance of software function, performance and support with software design. The chosen reusable components and newly developed components are integrated into system according to the reviewed software architecture and design.

(6) Software validation testing. Software validation testing is carried through to improve and validate the accordance of software function, performance and support with software assignment

and requirements, and system joint tests are implemented if demanded or needed. And necessary training and testing is actualized to solve possible problems and ensure correct system configuration, and meantime software reuse components are validated to meet the system needs and referred to software reuse library.

(7) Software delivery. Software product and related operation materials are delivered to users after software validation. Software system is rightly run by well trained persons; and software maintenance works are carried out according to development procedure if there is fateful revision. Software reuse evaluation is implemented along with the summarization of software development and report on software operation. Software reuse is generally evaluated by the weight addition of sub-characteristics such as mutual operation, customization, maturity, comprehensible, complexity, environment independent and software scale.

Software Reuse Framework

Software reuse framework comprises a lot of basic elements of figured component units, and it can quickly and effectively construct one applied system that is well adapted for requirement diversification, and gradually accumulate software assets by means of component library based on the perfect combination of component technology, bus technology and visualized development technology [5][6]. Software reuse framework provides a visual development and management platform for the whole software system based upon distributed computation, which is shown in Figure 2.

(1) Application layer. Application layer aims at particular business requirements, implements a set of business logics and provides a series of functions and services by means of the integration and configuration of various types of components and services.

(2) Data layer. Service data object supplies one standard data access interface and operation model that is irrelevant to related realization technologies, and includes data source, data access service and data chart.

(3) Service layer. Service layer comprises development environment, management environment, operation environment and related platform services of transfer, situation and security that contribute to the data sharing and mutual manipulation. And application treatments can shift to the related environment or refer to particular services.

(4) Infrastructure layer. Infrastructure layer comprises infrastructure resources such as network, hardware, bus and etc, and contributes to the dynamic extension and agile creation of electronic information system.

On the basis of the software reuse framework, software reuse platform can be set up to constitute integrated management system of component library that consists of basic components, current business components and other components developed by the third party. It provides the support for the life cycle of component production, operation and management that contains component definition, release, access and operation.

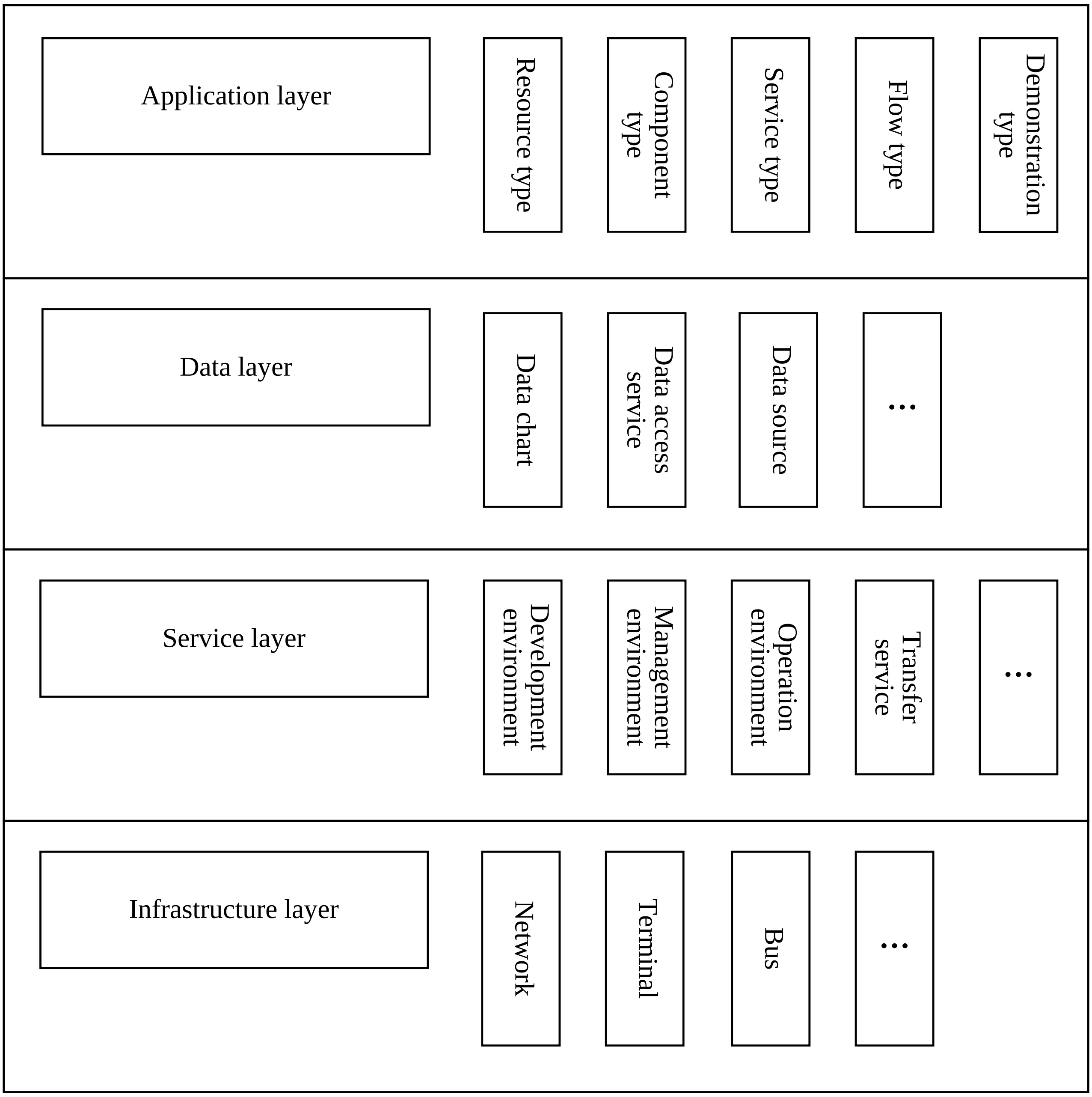

Fig. 2. Software reuse framework

Platform Application Mode

As regards the practical application, various types of components are properly obtained and integrated to realize a whole applied system by means of platform environments [7][8][9]. The functional parts mainly actualized related platform functions, which include component development environment, management environment, operation environment and component library that are shown in Figure 3.

(1) Component development environment provides development and debugging tools for the construction of reusable components and improvement of component library.

(2) Component library consists of basic components, current business components and specific components developed by related units.

(3) Component management environment provides the support and services for the life cycle of all types of components, which contain the visual definition, installment, release, operation engine and component access and supervision.

(4) Component operation environment provides visualized definition tools of business flow to

sustain rule setup and automation control of activities, sub-flows and workflow, and testing and configuration management tools to carry out component testing and version control.

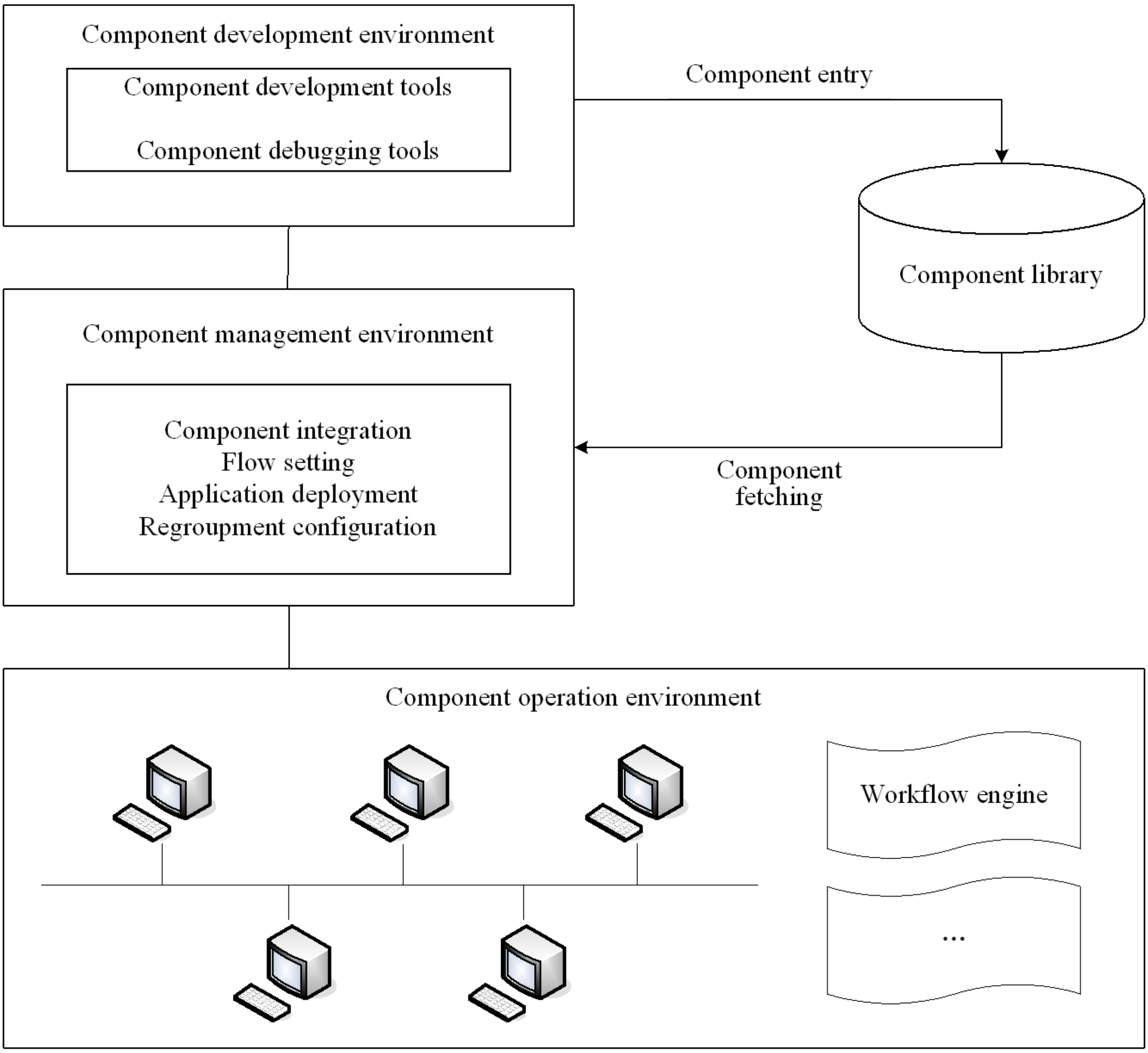
Component development environment
Component development tools
Component debugging tools
Component entry
Component library
Component management environment
Component integration
Flow setting
Application deployment
Regroupment configuration
Component fetching
Component operation environment
Workflow engine
...

Fig. 3. Platform application mode of software reuse

In addition, the related components from component library can be installed to form business logic and system application, and it can agilely answer to requirement change. Users' requirements and opinions should be fast and truly comprehended via wide communication with respect to software components, so that we can perform pertinent software and component design and realization; independent testing is helpful and valuable to the agility of software and component development, and the testing completeness and specialty technology are important and necessary for the elimination of software problems.

Conclusion

In the light of service oriented architecture technology, we establish software reuse framework, set up the software reuse and integration mechanism, and actualize applicable reuse platform, so as to agilely constitute applied systems in answer to requirement variation; moreover, we should further advance the construction of reuse component library, and properly carry out domain analysis and collectivity scheme, so that system agility and applicability could be remarkably improved with respect to system demonstration, overall scheme and development.

Acknowledgement

I am grateful to the specialists and secretaries for their insight and guidance, understanding and patience, and they are a constant source of support and encouragement.

References

[1] Fuqing Yang. Software production technology base on reuse [J]. China Science, 2001 (4) 363-371.

[2] Zhang W, Mei H. Feature-oriented software reuse technology-state of the art [J]. China Science Bulletin, 2014 (59) 21-42.

[3] Yasha Wang. Manual of software reuse standards [M]. Beijing: Electronic Industry Press, 2004.

[4] Yue Qiu. The research and realization of application integration policy based on SOA [J]. Computer Times, 2008 (2) 40-42.

[5] Hongyu Yao, Suning Tian: Cloud Computation [M]. Beijing: Electronic Industry Press, 2013.

[6] Xiaohui Li. Summarization of Research and Application of Cloud Computation Technology [J]. Electronic Measurement Technology, 2011 (34) 7-9.

[7] Xiaoqin Li, Hong Li. Application of Reuse Technique to Software Engineering [J]. Computer Development and Application, 2011 (2) 58-59.

[8] Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software [M]. Reading: Addison-Wesley, 1995.

[9] Li Ya. Research on the Application of Software Reuse Technology Based on Components [D]. Chongqing University, 2007.