A framework to support flexible application collaboration in cloud computing

Meng Xu, Qingzhong Li*, Lizhen Cui

School of Computer Science and Technology, Shandong University, China

Shandong Provincial Key Laboratory of Software Engineering, China

Received 1 October 2014, www.cmnt.lv

Abstract

With the development of Internet, software has more and more collaborative trends. Collaboration application is becoming a new hotspot. Development of collaborative application process is complicated and need many technologies to realize it. It is difficult for an ISV (Independent Software Vendor) or a business department to provide all application modules of a whole collaboration application. Moreover it is difficult to integrate heterogeneous application module developed by different organizations (ISV or business department). According to the problems mentioned above, this paper proposes a PaaS platform framework to build collaboration application dynamically and flexibly. Such PaaS platform can provide the ability of application delivery deployment for organizations and provide the ability of collaboration application customization for end users.

Keywords: cloud computing, PaaS, collaboration application, process

1 Introduction

With the development of Internet, software has more and more collaborative trends. Collaboration application is becoming a new hotspot. Industry-specific functionality software has huge demand for complex collaborative applications. Collaboration is based on existing business organizations, through information exchange between applications and resource sharing, to achieve a unified business objective. For example, in the field of e-commerce, collaboration between different organizations is very obvious. For an e-commerce application collaboration system, it needs to achieve integration of many e-commerce applications, like online shop system, electronic payment systems, logistics systems. Resources and services that belong to different organizations and business execution is cross-organizational collaboration. It needs a lot of the underlying implementation technology to building such a huge complex process information systems. Currently it is lack of support for collaborative applications dynamically build support platform.

PaaS (Platform as a Service) [10] as a cloud computing [1, 2] service model that provides a new kind of software development, delivery, deployment and usage patterns, but also bring new technical support for designing, developing large-scale network collaborative applications. However, most studies on the PaaS focused on the delivery, deployment and on-demand of single application at present. In this paper, we propose a PaaS to support dynamically building collaboration applications and scheduling methods. The platform is notable feature is support dynamically building collaborative applications.

The key business capabilities of platform include:

1) Application delivery management. Platform supports rapid development, delivery and deployment of software applications with standard of platform. Software delivered by organizations will be resolved, reconstruction of cloud services which can be customized and deployed in the application runtime environment for the users by platform.

2) Collaboration applications customizing. Platform provides rapid collaboration applications customization, development capabilities. Users through leasing rather than invest to build collaboration applications. Moreover, collaboration applications can be customized to meet users' business needs.

3) Collaboration application scheduling. Platform can schedule customized collaborative applications.

4) Cross-cloud services called. Platform supports organization register local service to the platform to achieve cross-cloud service call.

By supporting dynamically building collaboration applications, users only need care about their own business logic implementation, without considering the complex collaboration in ecological changes. On the other hand, users can choose on-demand cloud services by dynamically building collaboration platform. Users do not need a separate investment to build its business collaboration applications. On the contrary users can quickly build the characteristics of the enterprise business processes only using a variety of software services provided by others. The platform can effectively use resources, reduce costs, and ultimately improve the enterprise's core competitiveness to meet the user demand

Corresponding author e-mail: lqz@sdu.edu.cn on business collaboration by supporting rapid collaboration applications building.

The following e-commerce example illustrates users to build collaboration application using PaaS. Entire e-commerce collaboration applications are so complex that no one organization can provide complete application. On the other hand, it is difficult for SMEs to have the economic strength to invest in construction of such a huge information system. Various types of e-commerce applications (such as the portal system, online payment systems, online e-commerce logistics system core applications), which are developed based PaaS can be delivered, deployed to the platform. E-commerce customers can browse, leasing, custom e-commerce software services, e-business collaboration applications quickly customize and development.

The rest of the paper is structured as follows: Related work is discussed in Section 2. Then section 3 describes the architecture of PaaS to support collaboration applications dynamically building. The method of dynamically building collaboration applications will be presented in Section 4. And Section 5 will show the customized approach of collaboration applications based PaaS. Finally Section 6 concludes.

2 Related works

PaaS is an Internet-based cloud computing service model that provides a novel software design, development, delivery, deployment and usage patterns. PaaS provides all the necessary resources, including virtual servers, storage, databases, and so on for software developers to build and deploy applications. PaaS hosted on web-based system through the application development platform, providing end to end development environment, or in some cases, all local online development environment [3].

Some PaaS currently exist, and different PaaS are technically differentiated:

1) One class is to provide application development and deployment container, which provide necessary resources to applications. A typical representative is AWS Elastic Beanstalk [7], on which developers can quickly deploy Java applications to Amazon's machine.

2) Another class is to provide an open API for third-party. A typical representative is Forec.com [5], which provides the necessary development tools and application services, so that software developers can use it to quickly build scalable applications and do not need to purchase, install, configure and manage hardware and software.

3) The last class is to provide the ability to generate the application's configuration. A typical representative is LongJump [9].

However, these PaaS less concerned about the collaboration of software services on the platform at present. Literature [11] presents a Service Delivery Platform for the telecommunications industry by using PaaS technology. It achieves develop value-added telecommunications services through the platform services and registered on the platform assembly of external services. Literature [12] studied how to improve information integration framework petrochemical execution efficiency through the cloud computing service model.

3 Architecture

The different software services are assembled together to form collaboration applications, in fact, is the process of software reuse. Researches show that software reuse in a specific domain is easier to achieve [13, 14]. With the same business objectives software delivered by different organizations typically form collaboration applications in specific domain. "Domain" refers to areas covered by a group application with similar functions [4]. For example, e-commerce or tax applications will be reflected in the relevant software applications and features in common. PaaS proposed in this paper can dynamically build collaboration applications in different fields. The application delivered to PaaS should be specified an area of software applications, so platform can dynamically build collaboration applications in specific areas of software services. During collaboration application development, the characteristics of the domain can be used, such as a common system architecture, which enhances the efficiency of collaboration application development.

PaaS which support dynamically building collaboration applications needs to provide application delivery, collaboration applications rapidly customize, instant deployment, external integration and other key operational capacity. By using the web-based development tools, dynamically build collaboration mechanism provided by the platform, in accordance with their specific business collaboration needs, users can quickly customize collaboration application system including personalized interface, business processes and data models. Moreover users do not need to focus on application deployment details and can ensure that applications can run stable.

The feature of PaaS proposed in this paper is the ability of dynamic building the collaboration of applications which is provided by different organizations within a specific domain and achieve personalized customization needs of users. Figure 1 shows the overall architecture of PaaS, including application delivery module, collaboration as a service, application customization module, resource management module, application runtime platforms, database as a service and PaaS operations management system. The following describes the role of each module.

3.1 APPLICATION DELIVERY MODULE

Application delivery module provides a Web-based application packaging, analytical, deployment, maintenance function and the necessary support of delivering the software for the ISV. It can achieve applications developed by ISV from the traditional development environment to the platform. ISV can encapsulate the business logic, user interface, data model, and collaboration constraint in a compressed package for submission to the platform. The software developed by ISV can be expressed as App = <UI,BL,D,C>, where UI represents the application's user interface, including the interface of the function menu, the form layout, the optional background and Logo; BL is the core business logic of the application, which can be further divided into modules that can be customized by the user; D represents the data model of the application, including data objects and various data items of data objects; C represents the application collaboration constraints. These constraints can express which types of applications can be collaboration.

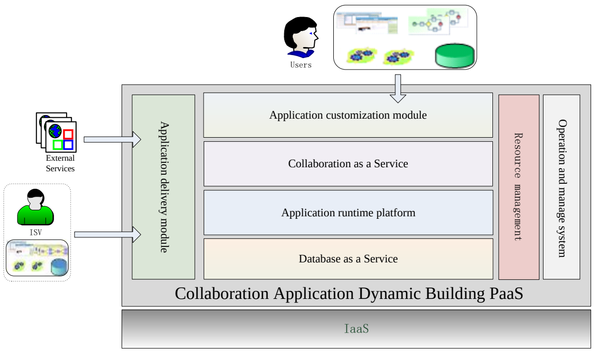
Users
External Services
ISV
Application delivery module
Application customization module
Collaboration as a Service
Application runtime platform
Database as a Service
Resource management
Operation and manage system
Collaboration Application Dynamic Building PaaS
IaaS

FIGURE 1 Overall architecture

PaaS will be responsible for the operation and maintenance work. Application Delivery module is responsible for parsing applications developed in accordance with the development standard and breaks it into the user interface, business logic, data models, and collaboration constraints. Then a service is built to enable users customizing collaboration applications. On the addition, PaaS can also accept registering local service on PaaS to enable cross-cloud service call capabilities. Use registration, organization needs maintenance its own software and infrastructure.

3.2 APPLICATION CUSTOMIZATION MODULE

Users can customize a single application on platform, including user interface, business logic, and data model and other aspects. Collaboration application customization module is shown in Figure 2. It includes UI customization module, collaboration application customization module, data customization module, cross-cloud service invocation interface module, custom verify module and custom isolation module. UI customization module supports the customization of menu, form layout, UI themes, interface colours, LOGO and other presentation layer.

Collaboration application customization module can implement customization on individual service and collaboration applications. Since collaboration constraints of each application are stored, platform will search, match, association, and recommend some abstract collaboration applications for users according to collaboration constraints.

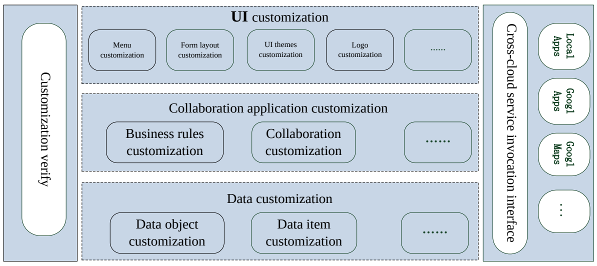

FIGURE 2 Application customization modules

Data customization module is responsible for the customization of data model, data object, and data item. It provides the ability to add, delete, edit and other customized functions of data objects and data items for tenants. Cross-cloud service call interface module is responsible for integrating a variety of cross-cloud applications. Custom verify module is responsible for verifying the user's customization results to ensure that the results did not violate the customization constraints.

3.3 COLLABORATION AS A SERVICE

The software delivered to PaaS is parsed and rebuilt as a standard cloud services by the application delivery module. There may be collaboration relationship between these services from different organizations. Collaboration as a service can discover the collaboration of services in services pool according to the collaboration constraints. Then it will link these services to form an abstract collaboration application. The abstract collaboration application refers each service node of abstract collaborative applications represents a class of services which have similar functions. For any service node, there may be many services from different organizations which can be chosen. The Collaboration as a Service is shown in Figure 3.

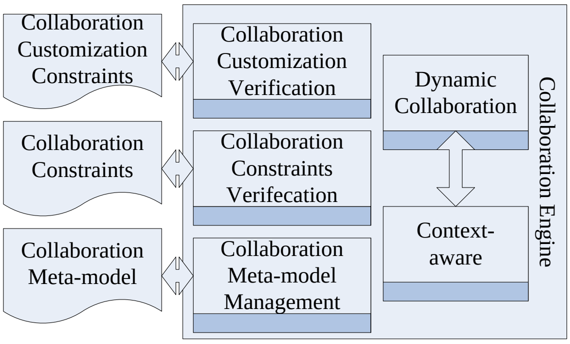
Collaboration Customization Constraints
Collaboration Constraints
Collaboration Meta-model
Collaboration Customization Verification
Collaboration Constraints Verifecation
Collaboration Meta-model Management
Dynamic Collaboration
Context-aware
Collaboration Engine

FIGURE 3 Collaboration as a service

3.4 APPLICATION RUNTIME PLATFORMS

Application runtime platform is the runtime environment, including UI engine, data engine and business process engine. These engines can interactive with application deliver module, application customization module, PaaS operations management system. The collaboration applications generated by collaboration application customization module are deployed to the application running platform to provide services for users.

UI engine is responsible for UI presentation. It can rebuild, adjusted and displayed UI interface on the terminal. Data engine is used to parse data model of applications. Business process engine can parse and instantiate process of customized applications. It is responsible for creating and scheduling process instances, and process monitoring and exception handling. In addition, the application runtime platform can also call a Web service across cloud.

3.5 DATABASE AS A SERVICE

Database as a Service is responsible for managing data in the platform which include application data, meta-data of application, collaboration data and other data. Database as a Service can provide a uniform data model to all kinds of application which makes the manipulation of data more efficiently. On the other hand, applications can access their own data through different API. It is used as if only one was using the service. For example, application A can use MySQL database, while application B may use MongoDB.

3.6 RESOURCE MANAGEMENT MODULE

Resource management module can manage services according to its domain. The services which belonging to the same domain will be unified management, which will be beneficial for collaboration as a service dynamically building collaboration applications according to different domains. Services catalogue is also a part of it. The description of software delivered or registered to platform, including functional and non-functional information, will be in services catalogue. Services catalogue is available for user to find services satisfied his requirements. Besides resource management module is also responsible for data management and maintenance.

3.7 PAAS OPERATIONS MANAGEMENT SYSTEM

PaaS operations management system is used to manage the registration of tenants, as well as tenant directory structure, indexes, service access management. It is also used to coordinate between PaaS and tenants about QoS assurance. SLA requirements for different platforms QoS metrics to quantify dynamically adjusted based on quantitative indicators of shared resources, service monitoring, when the real understanding of custom applications running conditions and usage. Ensure that the platform's reliability, stability and robust to ensure tenants to meet SLA requirements. SLA-based resource dynamic adjustment strategy will meet two objectives: first, the SLA requirements for high-priority users will get the quality of service assurance; two SLA requirements is to ensure a high quality of service under the premise of the user, try to take care of low SLA requirements of users quality of service.

4 Building flexible application collaboration

Dynamically building collaboration applications rely on semantic rules. The rules of semantic description will be given in this section. Applications delivered by different organizations should follow these rules. When an application is delivered to PaaS, the application will be classified according to semantic information, and dynamically discover, associate, and build abstract collaboration applications according to the collaboration rules of the applications. The main roles of the semantic model are: first, it is the application development standards; second, it makes dynamically build collaboration applications possible.

4.1 BUILDING ABSTRACT COLLABORATION

An industry collaboration application will have its regular logic process models, such as e-commerce, collaboration applications in this area will involve electronic transactions, online payment, online logistics systems, etc. And the entire collaboration model is basically fixed. Differences between collaboration applications are also in framework of collaboration models. So an analysis can be made by domain experts to extract the business collaboration process to obtain collaboration application model for a specific domain. In addition, as more and more applications are delivered to platform, the potential collaboration between applications will be auto-discovery, association to form a new collaboration application model, or add new applications to the existing collaboration model. Users can customize the collaboration model according to their own needs.

Definition 1: Each service node of collaboration model represents one class of services but not a specific service instance in which all services have similar functions. Each service class contains many service instances which can achieve this functionality. The collaboration model is defined as BP = <SC,E>, where:

1) SC = {sc1, sc2, ..., scn} is set of the service classes, sc1 .... scn are the n classes of services that participate in abstract collaboration applications.

2) $E \subseteq SC \times SC$ is the set of connections between the service classes, in which $X \in E$ is the structure control operator. A number of basic services can be combined into a service workflow through some control structures. The control structure operators used by workflow are sequence, selection, cycle, and parallel.

Definition 2: The service class is a class services with same or similar function. Any service in the class which can meet the performance requirements of a specific application can be candidate service as a service node of abstract collaboration application.

Definition 3: Collaboration constraints C is defined as C = <Sd,F,Q,Dom>, where Sd represents the basic definition and description of the application, including the service ID, a basic description of the application name. F represents the functional description of the application, including application interface parameters, preconditions and postconditions; Q is the non-functional attributes descriptions of the service, including time, cost, reputation, etc. Dom represents an application domain.

Users can cut, configure and build abstract collaboration applications which meet their own needs based on collaboration model. After building abstract collaboration applications, the next step is to bind service. The candidate services for each service node will be sorted according to their function by platform. Users can choose service to meet their non-functional requirements. Non-functional requirements include prices, response time, etc.

4.2 CUSTOMIZING COLLABORATION

Users can customize the application from the user interface, business logic, business process and data models four levels. This article focuses on the process logic customization of the collaboration application, i.e. users based on the abstract collaboration applications, according to their needs customize personalized collaboration applications. Platform provides the following custom methods:

Users select one collaboration application model, delete unnecessary path and change the service relationship based on this model. For example, for e-commerce collaborative applications, the user can adjust according to demand whether pay after shipping or shipping after pay. Eventually the abstract collaboration application is formed. Each service node of abstract collaboration application corresponds to many service instance, user can choose one service instance which meet his needs. For example, although many organizations have developed electronic payment applications, but there are some differences, user can choose on-demand according to their actual business conditions.

When there are a lot of collaboration models on platform, it is very difficult that users make choice manually. So a recommendation algorithm is proposed to quickly select the collaboration model.

Definition 4: Let ReqS is the services set of collaboration applications, Cm is a domain collaboration model, CmS is services included in the Cm , if ∀si∈ReqS, then si∈CmS, called ReqS match with Cms.

Definition 5: If ReqS match with CmS, and |ReqS| = |CmS|, called ReqS exactly match with CmS.

From the above definition, as long as ReqS match with CmS, the collaboration model will contain all functional requirements of user. But there may be some unwanted features. So the best case is the user needs and collaboration model exactly matches. If there is no exactly match, then find out the collaboration model with minimum difference.

Definition 6: Let C(ReqS, CmS) = $\frac{|\{si \mid (si \in \mathrm{Re}qS) \wedge (si \in CmS)\}|}{|\mathrm{Re}qS|}$ as the degree of collaboration model CmS cover user needs ReqS.

Definition 7: Let R(ReqS, CmS) = $\frac{|CmS - \{si \mid (si \in \mathrm{Re}qS) \wedge (si \in CmS)\}|}{|\mathrm{Re}qS|}$ as the degree of collaboration model CmS redundant of user needs ReqS.

Algorithm 1 recommendation algorithm:

```
Input: ReqS, userCov, userRed
Output: M

Ms= Ø //set candidate model null
for each m in M
  cov= C(ReqS, m)
  red =R(ReqS, m)
  If(cov>=userCov ∧ red<=userRed)
  M.add(m);
end for
Return M;
```

5 Scheduling Collaboration and Simulation

5.1 SCHEDULING COLLABORATION

Collaboration applications scheduling has two layers. The first layer maps the abstract collaboration application to the executable collaboration application. The second layer schedules the task to the appropriate computing nodes according to load balance.

Users customize the personalized collaboration applications based on the abstract collaboration applications. Then customized results are submitted, and will be scheduled by platform. The collaboration application scheduling model is shown in Figure 4.

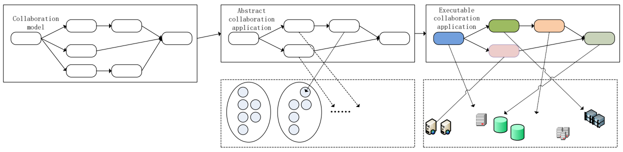
Collaboration model
Abstract collaboration application
Executable collaboration application
......

FIGURE 4 Collaboration application scheduling model

The first layer scheduling algorithm:

1) Process engine first reads the user's personalized needs, then the abstract collaboration is cut into customized collaboration application, i.e. removing the service node which user does not need.

2) The abstract customized collaboration applications bind services. The binding process is to find the corresponding services set (service instances set) based on the type of service. Then the services which are not satisfied QoS are deleted according to the user's QoS constraints.

3) After the above calculation, the service nodes of abstract collaboration application are mapped to specific service instance.

4) After executable collaboration applications generated, services is called and platform will monitor the execution of collaboration applications.

Taking into account the load balance, a service will not only deploy on one server, but have several copies on multiple servers. Then an executable collaboration application is scheduled, the appropriate server should be selected to make server load balancing. PaaS will receive a lot of request to execute applications the in a period of time, so multiple collaboration applications jointly scheduling should be considered. The strategy proposed in this paper is: first all executable services (execution condition is satisfied) is inserted into an execution queue according to the arrival time. Then try to do the following steps: a) remove the first task; b) schedule this task to a most appropriate server; c) If there is no appropriate server, this task is inserted into the tail; d) When the task is completed, notify scheduler task execution is completed.

Definition 8: The server load function is defined as $L(s) = \frac{C\max(s) - C_{cur}(s)}{C\max(s)}$, where $C\max(s)$ is max load of server, i.e. the maximum execution server capacity, $C_{cur}(s)$ is the server's current load.

If the current server is idle, that is $C_{cur}(s) = 0$, then $L(s) = 1$. If the current server load reaches the maximum value, that is $C_{cur}(s) = C\max(s)$, then $L(s) = 0$. So the second layer of collaboration applications scheduling algorithm is as follows:

1) For an executable collaboration application, all executable tasks will be inserted into a queue.

2) Take out each task, assign it to execution node. Execution node selection is based on its load function value. The value of the minimum execution node is selected. If all nodes load are full, i.e. L (s) <= 0, then this task is inserted into the tail.

3) Repeat steps 1 and 2 until the queue is empty.

5.2 SIMULATION

To verify the efficiency and scalability of the framework, an experiment is conducted. Five different test sets are used in the experiment. The collaboration time and result are recorded and shown in Table 1. If the collaboration which is generated dynamically by the framework in this paper can be executed and in accordance with the requirements, we call the collaboration is correct. And the accuracy in Table 1 is the correct rate of the collaborations. From Table 1, we can see that the execute time increases with the number of services linearly and the accuracy is always 100%.So we can conclude that the framework propose in this paper is effective and scalable.

TABLE 1 Collaboration time and accuracy

Test Set	Number of services	Time(ms)	Accuracy
Test Set 1	800	35	100%
Test Set 2	1000	52	100%
Test Set 3	400	116	100%
Test Set 4	1500	528	100%
Test Set 5	4000	1860	100%

6 Conclusion

Based on IT development and the demand of business collaboration, a PaaS for business collaboration is proposed that supports business collaboration applications quickly build and deployment. First, the platform provides a new application delivery model. The application can be parsed and rebuilt the services for the user customization. Secondly, platform provides automatic matching technology. The collaboration relationship between services can be matched and automatically generates business collaboration model. It simplifies the process of collaboration application customization. Moreover, the platform provides collaboration applications for deploying, runtime environment, including the UI engine, data engine and workflow engine.

Acknowledgments

This work has been supported by the National Key Technologies R&D Program under Grant No. 2012BAH54F01; the National Natural Science Foundation of China under Grant No. 61003253, No. 61272241, No. 61303085; the Natural Science Foundation of Shandong Province of China under Grant No. ZR2010FM031, ZR2010FQ010; Science and Technology Development Plan Project of Shandong Province under Grant No. 2012GGX10134; Independent Innovation Foundation of Shandong University under Grant No. 2012TS075, No. 2012TS074; Shandong Province Independent Innovation Major Special Project under Grant No.2013CXC30201.

References

[1] Buyya T, Yeo C S, Venugopal S, Broberg J, Brandic I 2009 Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility *Future Generation Computer Systems* Elsevier Science Amsterdam **25**(6) 599-616

[2] Armbrust M 2009 Above the Clouds: A Berkeley View of Cloud Computing *EECS Department University of California* Berkeley

[3] Lawton G 2008 Developing Software Online with Platform-as-a-Service Technology *Computer* **41**(6) 13–5

[4] Zhao J-F, Xie B, Zhang L, Yang F-Q 2005 A Web Services Composition Method Supporting Domain Feature *Chinese Journal of Computers* **28**(4) 732-8

[5] Salesforce *Force.com: The leading cloud platform for business apps* http://www.salesforce.com/platform/

[6] AppEngine http://code.google.com/intl/EN/appengine/

[7] Elasticbeanstalk http://aws.amazon.com/elasticbeanstalk/

[8] Anstett T, Leymann F, Mietzner R, Strauch S 2009 Towards BPEL in the Cloud: exploiting different delivery models for the execution of business processes *2009 World Conference on Services – I* 670-7

[9] Longjump, http://longjump.com/

[10] Zhou Y C, Liu X P, Wang X N, Xue L 2010 Business Process Centric Platform-as-a-Service Model and Technologies for Cloud Enabled Industry Solutions *2010 IEEE 3rd International Conference on*

Cloud Computing (CLOUD) 534-7

[11] Zhou Y C, Xue L, Liu X P, Wang X N, Liang X X, Sun C H 2010 Service Storm: A Self-Service Telecommunication Service Delivery Platform with Platform-as-a-Service Technology *2010 IEEE 6th World Congress on Services (Services – I)* 8-15

[12] Zhou Y C, Wang X N, Liu X P, Xue L, Liang S, Sun C H 2010 Enabled Integrated Information frameworks for chemical and petroleum industry *2010 IEEE 6th World Congress on Services (Services – I)* 1-7

[13] Prieto-Diaz R 1987 Domain analysis for reusability *Proceedings of COMPSAC'87* Tokyo Japan 23-9

[14] Mili H, Mili F, Mili A 1995 *IEEE Transactions on Software Engineering* **21**(6) 528-62

Authors

	Meng Xu, born in October, 1978 **Current position, grades**: PhD Student, Shandong University, Jinan, Shandong, China. **University studies**: master's degrees at Shandong University, 2010. **Scientific interests**: cloud computing, service computing. **Publications**: 10 papers.
	Qingzhong Li, born in 1965 **Current position, grades**: PhD, professor and doctoral supervisor at Shandong University, Jinan, Shandong, China. **University studies**: PhD degree at Institute of Computing Technology of Chinese Academy of Sciences, 2000. **Scientific interests**: data management, cloud computing, data integration. **Publications**: 60 Papers.
	Lizhen Cui, born in 1976 **Current position, grades**: PhD, associate professor and master supervisor at Shandong University, Jinan, Shandong, China. **University studies**: PhD degrees at Shandong University, 2006. **Scientific interests**: cloud and service computing. **Publications**: 30 papers.