

<http://ansinet.com/itj>

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSI*net*

Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Research of Configuration on Multi-tenant

Shengwen Zhang, Zhaohui Li and Xiaoxi Chen

Transportation Management College, Dalian Maritime University, Liaoning, (116026), China

Abstract: The SaaS (Software as a Service) application model comes into being along with the expansion and convention of the network platform and its core lies in Multi-tenant application. The new software model based SaaS has a great temptation for enterprises. The new software model eases the fund pressure of enterprises virtually so that enterprises can concentrate on their own business development. However, the large customers bring the issue of individual needs. The tenants from different enterprises put forward different demands according to their own internal management way and process. How to meet the individual needs of enterprises is the main content of this study. Present study takes Dalian ted talent port for example. It mainly researched personalized configuration on data, function and interface and gave a personalized configuration way, achieving Multi-tenant application and managing these applications through personalized configuration.

Key words: SaaS, multi-tenant, personalized configuration

INTRODUCTION

SaaS (Software as a Service) application model comes into being along with the wide use of Internet, its core application lies in realization of Multi-tenant. The development of Multi-Tenant application system bid farewell to traditional software development model, instead, it formed a new model that makes many enterprises hire a new system use case via Internet (Wei, 2009). This model allows users not to deploy application software in their own computers and servers and not to spend fund and effort maintaining and upgrading software which are charged by service providers. More importantly, it greatly reduces software development cost, hardware acquisition cost, training cost, system maintenance cost and so on. This model eases cost pressure invisibly so that enterprises can concentrate on their own business. Meanwhile, SaaS software operation environment is through a common Web browser platform. So tenants can receive software service and operate office business as long as they can connect to network.

For a large mature Multi-tenant application system in SaaS architecture, the large customers bring some problems about individual needs. Tenants from different enterprises will propose personalized needs about system function, data, process and interface according to their own internal management way and process (Shichao, 2010). SaaS architecture for the purpose of service must meet these diverse individual needs. In order to meet these individual needs of tenants, the shared use case must have personalized configuration which include background data configuration, application function

configuration, business process configuration, operation interface configuration. How to meet these needs of individual configuration, how to allow tenants to use the same use case without disorder about function and how to make tenants have a higher satisfaction are the core problems that need to be solved by Multi-tenant application system in SaaS architecture. Present study takes Dalian ted port as an example. It mainly researched personalized configuration on user data in the database layer, system function in the Web logical layer, operation interface in the presentation layer, meeting the needs of different tenants.

THEORY RESEARCH ABOUT MULTI-TENANT APPLICATION SYSTEM

Personalized configuration about user data: When Multi-tenant applications are used, some tenants will always need to save some data which is redundant to other tenants. But it is difficult to convince customers not to have such a demand. This data is known as “Extension Data”. Obviously, the extension data that different customers want to store is completely different. However, in the multi-tenant SaaS application, all of the tenants use the same application instance. There are three common solutions to solve the configuration of extension data: Customize fields; Pre-allocation fields; Expansion association table (Wei, 2009; Aulbach *et al.*, 2008).

Customize fields: Customize fields are a traditional data structure implementation. When the tenant's demand for

data is changing, we increase the corresponding fields in the data table. This way of extension data is the simplest. As long as the tenants have demand for the fields, we can directly change or expand the corresponding fields in the original table. If the database is the exclusive, the way of additional fields can be accepted and will not affect other tenants. But if the Multi-tenant application system allows tenants to add customize fields, then these fields will be too much to count. Data structure will be seriously damaged by large amount of redundant data, affecting the use efficiency of other tenants.

Pre-allocation fields: Pre-allocation field is assigning a certain number of pre-expansion fields in the table. When tenants need expand fields, they can use these pre-allocation fields. When tenants have additional requirements they can use these pre-allocation fields. However, it will also have a big problem. What is the meaning of pre-allocation fields for current tenant and what is the meaning of the same pre-allocation field to different tenants? If there is one pre-allocation field, tenant needs two fields. How to solve this problem? If pre-allocation fields are designed too much, it will result in database expansion and it also causes a large number of meaningless, which leads plenty of storage space is wasted.

Expansion association table: Expansion association table is adding a table to store fields which need to be extended and adding a table to store extended data. The three tables meet the different needs of different extension need from different tenants. Expansion association table solves user-defined extension fields in a very flexible method. When tenant adds a customize field, the extending-fields table will be stored an extension field. Mean while the row corresponding with the original table will add an expansion data to expansion-data table. It will not cause a waste of database space and demand for data expansion of tenants will be well meet.

Comparison of three schemes: Due to the three solutions to realize data configuration: Customize fields; Pre-assigned fields; Expansion association table. There are many differences in space complexity, space utilization, the flexibility and scalability. Detailed comparison is shown in Table 1.

By comparison of the options above, implementation complexity of expansion association table is higher than others. But, in general, the solution meets the requirements of Multi-Tenant application. Application examples in this study use expansion association table to solve data configuration.

Personalized configuration about system function: Multi-tenant system in SaaS architecture advocates “on-demand use, on-demand pay”. That is, it allows tenants to have a choice to buy the function according to their need. Different tenants can use a different combination of functions. This can solve the problem that the needs of different tenants do not completely overlap. It adapts the functional requirement differences between tenants in different industries in a greater degree. In addition, even for the same tenant, at different times, they may also have different needs. Allowing tenants to choose a different function set and pay for it, to a certain extent, it reduces cost pressures for the corporate. Personalized Configuration about System function solves this program well so that different tenants use different system functions in the same case. So how to realize the function that tenants use different functions online.

No matter how complex and how large the system is, it is composed by a variety of different functions for users. If all the functions of the system are broken down into basic functions of the smallest unit, these micro basic functional units are easy to handle and it make it easier for tenants to operate function configuration.

System function division: Functional decomposition is to decompose the system into the most basic atomic function of the relative independence. Functional decomposition should follow the following the main principles below (Wei, 2009; Jansen *et al.*, 2010):

- Each feature is valuable
- The function after the breakdown can not be subdivided, that is to be minimized
- The function after the breakdown can not overlap each other
- The function after the breakdown does not allow circular dependencies
- Function of the whole system is complete

Design of function pack: From the results of the above function division, we can find that not all of the atomic functions can be used alone and there are some dependencies between functions. It is an extremely complex matter to configure the atomic functions directly. So consider the interdependent atomic function put together.

Table 1: Comparison of data configuration

Comparison items	Customize fields	Pre-assigned fields	Expansion association table
Scalability	Low	Middle	High
Flexibility	Low	Middle	High
Implementation complexity	Low	Middle	High
Space utilization	Low	Middle	High

Personalized configuration of operation interface: No matter user data or system function, the most direct configurable feeling is reflected in interface. Good user experience will improve user satisfaction leasing the system. Personalized configuration of operation interface mainly reflects in two aspects: the content of operation page and personalization of menu.

Configuration of system menu: Different tenants can buy and use a collection of different functions within the system. But the same function menu for different users may have completely different names. In addition to name can be configured, the menu hierarchy and distribution can also be configured according to the requirements of the different configuration. In the configuration of system

menu, we need to consider the following questions below (Wei, 2009):

- One tenant has a menu
- A menu can be associated with a kind of function
- Constitute a tree structure, form on level menu structure
- The problem on the display order of the same level menu

In summary, we can get the class diagram shown as in Fig. 1 through simply designing the configuration content of system menus.

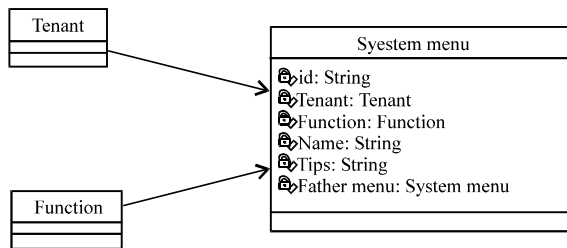


Fig. 1: Class diagram of configuration about system menu

Configuration of page elements: It is similar to system menu; the content on function pages is interface element, interacting between users and system. Different tenants may have different demands. Whatever the number of page elements is, the position of page elements, order of page elements or element meaning, tenants are likely to have some personalized need. In addition, user can define the extending data by themselves in the data configuration, so it is easier to use system for users by configuring elements on function pages. It's a common practice to use tag language description.

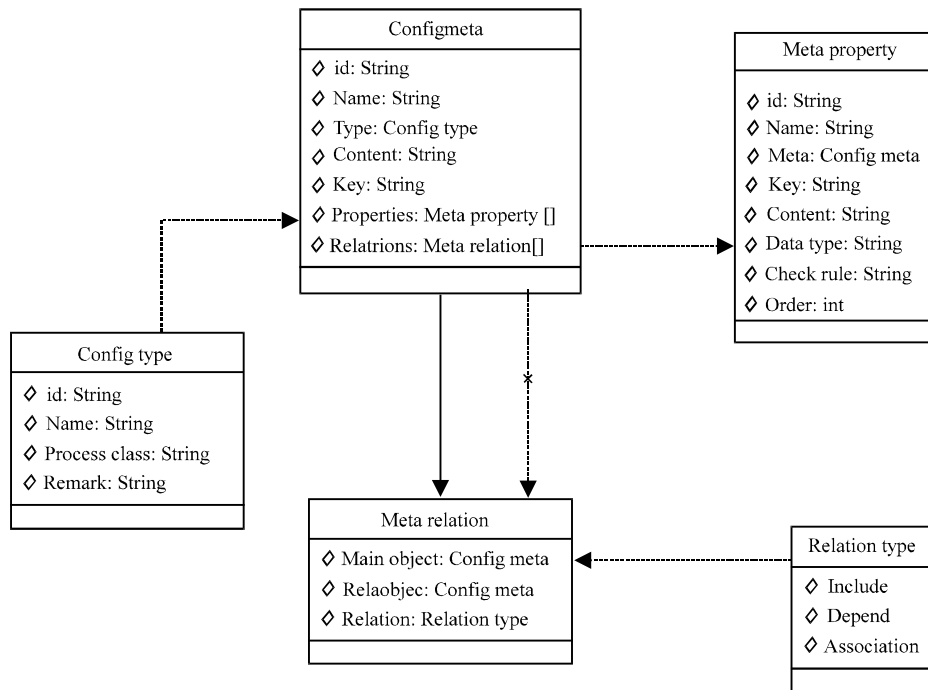


Fig. 2: Class Diagram of the configuration meta-data

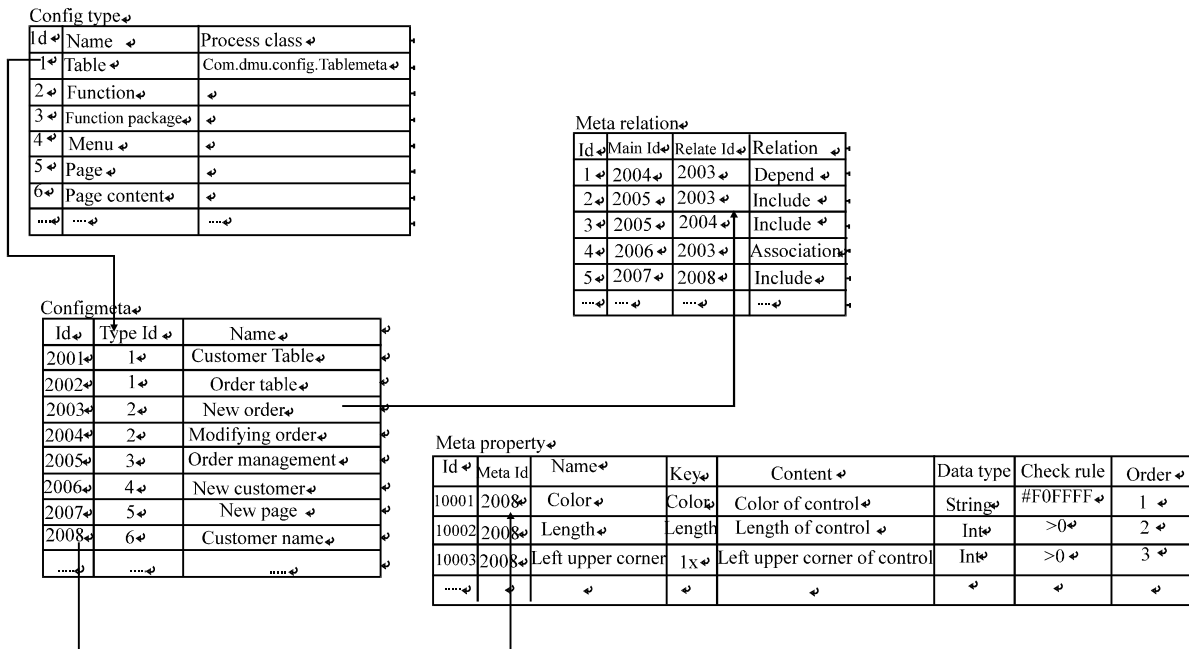


Fig. 3: Relationship of meta-data

Meta-data model of interface: The page elements in the users' operational interface may be different in terms of both the layout and the amount. It is complex to configure individually. Here, we use MDA (Shichao, 2010; Jansen *et al.*, 2010). (Model Driven Architecture) based on the idea of metadata management, to unify the management, including system function menu and configuration parameters of the elements shown in a page. The detail diagram of Meta-data Configuration is shown as Fig. 2.

In Fig. 2, the class ConfigMeta is used to indicate some common attributes of the points that are needed to be configured; the class ConfigType contains various configuration points; the class MetaProperty is the description of the extended attributes of configuration points. According to the class figure, corresponding diagram of meta-data configuration forms. The specific relationship between them is shown as Fig. 3.

In the Fig. 3, the Typed of "Customer Name" of No.2008 is 6. Dating back to Table ConfigType, we can find that the id of PageContent page element is. The control shown as "customer name" has three individual configuration requirements, namely (10001, color) the color configuration requirement; (10002, length) the length of the control configuration requirement; (10003, 1x) the configuration requirement of the position of the control from the upper left corner. By using the configuration metadata, configuration parameters can be managed together.

MULTI-TENANT APPLICATION SYSTEM PROJECT APPLICATION

The background of the case: Dalian ted talent port (www.taderhr.com) is a Multi-Tenant application in SaaS architecture (Lin and Han, 2010). The whole system involves service providers (Dalian Ted Company), service users (small and medium enterprises, enterprises providing indirect service), individual users. Service providers offer a variety of online service about HR. Including: training station, club, tool house, evaluation bar, shopping hall and headhunting port. Service users are divided into two types of users: Service indirect providers and service consumers, that is, service indirect providers can use online platform to post service provided by service providers. Service consumers purchase online service provided by service providers or service indirect providers. Individual users can not only purchase online service provided by service providers but they can also join the indirect providers to use service. Due to Multi-tenant model, different tenants have different function demands according to their own business. Therefore, the personalized configurable technology about Multi-tenant model is very necessary when the system is realized.

Configuration of user data: HR service provider's primary service is for human resource. Background management system gives service customers function modules about

personnel, payroll and report. Different service customers involve different industries, different departments and different positions. So it is necessary to let customers configure some basic setting fields. For example, different service customers are from different functional departments which should be configured by service customers. So the system should provide department setting page to add functional departments. If we carry out many to one between functional department field and service customers that is, we give one service customer a unique field-id, we can achieve personalized configuration about user data (Wei, 2009). Then service customer can save their own departments by setting functional departments. The system will directly load functional departments when management page is loaded which is selected by users. Realization ways are shown in Fig. 4.

In the above class diagram, AbsAnnoEntity, AbsAnnoCodedEntity, AbsAnnocodedNamedEntity are three abstract classes. The three classes define commonly basic fields of the whole system, including id, code, name and description. Class tenant (Tenant) and Class functional department (PostLevel) obtains the appropriate fields through inheritance Class AbsAnnocodedNamedEntity. In the class diagram, PostLevel has a tenant class which defines the ownership of the functional department. When the tenant handles department management business, he can get their own functional department by comparing Tenant-id and PostLevel-organizer.

Configuration of system function: Service provided by HR service providers involves many aspects. The service includes: training, HR measurement, HR tools and recruitment and many other features. But different service customers need different HR service according to their own business and different development stage. It is necessary to configure the system functional configuration for service customers.

Through analyzing the system functions, the function of the system is divided into atomic function and related atomic functions are combined with one function package. It is shown in Fig. 5.

A system functional configuration note only involves function selected by tenants but also involves system function permission using by different users (Jansen *et al.*, 2010). Therefore, configuration is particularly complex. However, the Lander is specific to a user, so we only link function node, user and permission. Class diagram about system function configuration is shown in Fig. 6.

In the above class diagram, class user is a tenant. Class FunctionNode is system function class. Class

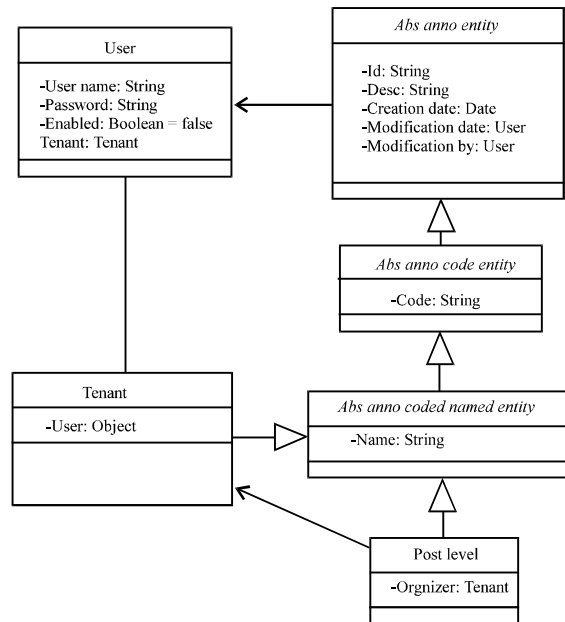


Fig. 4: Multi-tenant application class diagram of functional department

SystemResource is system resource class. Because system in B/S architecture adopts URL mapping relevant resource page, this system unifies URL relevant resource into Class SystemResource. And Class UserRole is for identifying each user group utilizing permission of system functional module, while Permission class defines user privileges under the system functional node (including: save, update, finish and validation, approve etc.). When the user logs in system, the system is loading user rights (UserRole) according to identification of users. System can load Permission corresponding to each privilege according to user rights-roles. Then system will load the corresponding system function node according to Permission. System function node is shown in Fig. 6 after users log in the system (Lin and Han, 2010). Different users have different system function nodes. When Lsander clicks links of function nodes, it could map URL to relevant page. But before mapping, system will judge whether the user has permission to operate data, if so, map to the request page, if not, map to a page that inform no privilege of the resource (Fei *et al.*, 2010). By means of that, we implement system functionality and user permission configuration.

By isolating user and function, service provider can configure different services to service customers, service customers will represent different services of using function module, so as to achieve function configuration. Different function display Figures of different users are shown in Fig. 7.

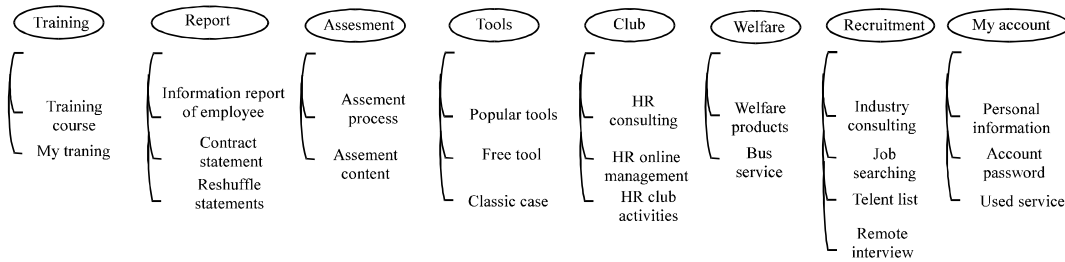


Fig. 5: Design chart about function package of system

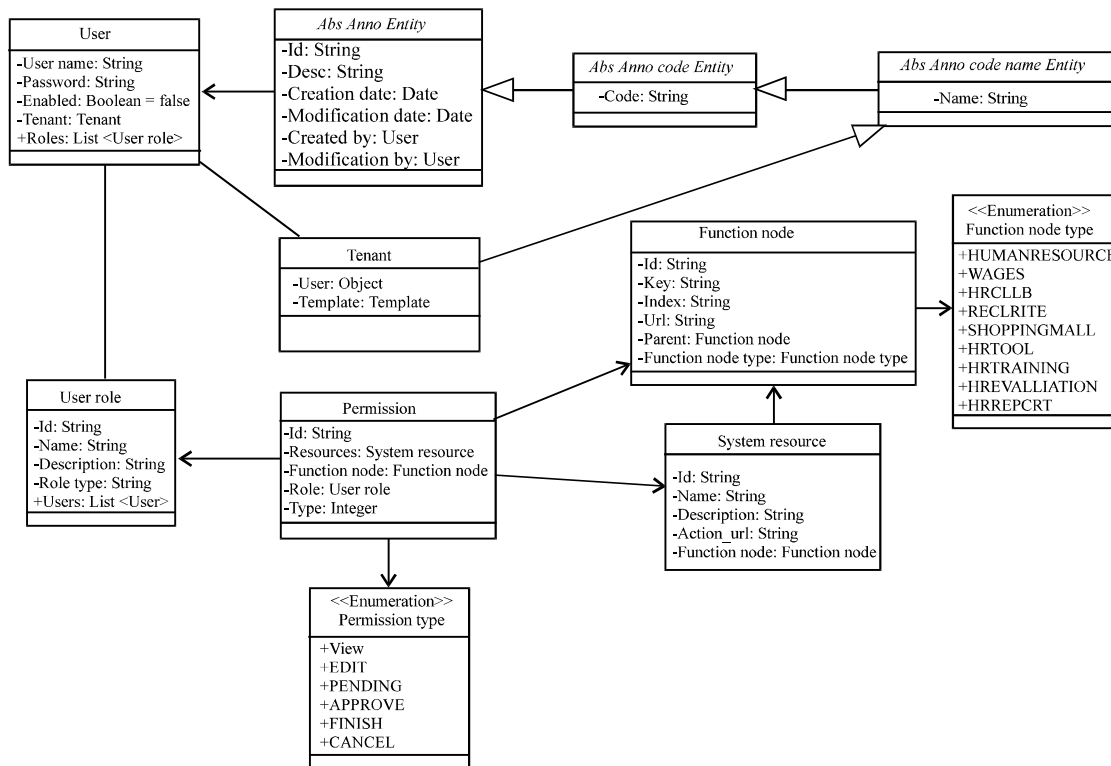


Fig. 6: Class diagram about system function configuration



Fig. 7: Different function configuration diagrams of different users

Personalized configuration of operation interface: Now, network has entered the stage of personal expression. Different users not only have different demands for use but also have different preferences for operation interface. In order to achieve individual needs of operation interface, the system takes template configuration. Users can select different operation interfaces provided by service providers so that interface is configured personally.

Service supplier provides initialization data of interface configuration which is different from personalized configuration about data user. Service providers proposed some demands on corresponding personalized templates for system developers. System

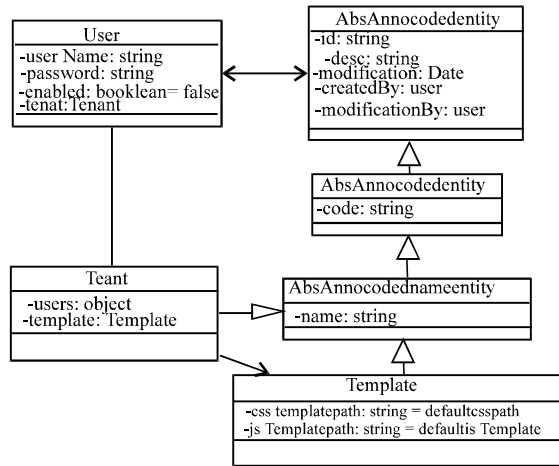


Fig. 8: Class diagram of operation interface template

developers develop system personalized operating configuration page about interface. Service providers configure interface through operating personalized configuration page to achieve personalized configuration of operation interface. In the realization process, System make template object-oriented and achieves a personalized configuration page by providing a different package configuration page to showing different paths (Fei *et al.*, 2010). The class diagram is shown in Fig. 8.

In the above class diagram, Template is a template class about operation interface, including personalized interface configuration path. This property is responsible for the system to locate and display the configuration template. Generally, the property includes different presentation of CSS and different gallery package. Js. TemplatePath property is responsible for personalized display of dynamic page operation shown by different templates. Each service provider (Tenant) can have a default Template. They set the template properties by choosing a different template to achieve personalized configuration of operation interface.

CONCLUSION

At present with the rapid development of Internet information age, variety factors make SaaS application model be superior to traditional software customization model. Multi-tenant application in SaaS architecture is a core issue. Present study based on this background

analyzes and studies personalized configuration of Multi-tenant application, which includes the following three aspects:

- Part 1 :** Personalized configuration of user data
- Part 2 :** Personalized configuration of system function
- Part 3 :** Personalized configuration of operation interface

The first part introduced the personalized configuration about user data. It mainly described the ways isolating and configuring user data by using databases and data structure. The second part introduced the personalized configuration about system function. It mainly described how to use sub-functional units to combine into application and check the permissions through user authentication center to meet customizable functions of different users. The third part introduced the concept of element model. It mainly described how to manage and deploy configuration parameters by the way of the structure of metadata and how to display and run system by a unified metadata platform.

REFERENCES

- Aulbach, S., T. Grust, D. Jacobs, A. Kemper and J. Rittinger, 2008. Multi-tenant databases for software as a service: Schema-mapping techniques. Proceedings of the International Conference on Management of Data SIGMOD'08, June 9-12, Vancouver, BC, Canada, pp: 1195-1201.
- Fei, S., W. Bing, W. Zhe and J. Xiaojuan, 2010. Research on laboratory information management system based on SaaS mode. Exp. Technol. Manage., 27: 97-100.
- Jansen, S., G.J. Houben and S. Brinkkemper, 2010. Customization realization in multi-tenant web applications: Case studies from the library sector. Web Eng. LNCS, 6189: 445-459.
- Lin, H.L. and Y.B. Han, 2010. Performance management for multi-tenant web applications. Chin. J. Comput., 33: 1881-1895.
- Shichao, J., 2010. Research of Configuration on Multi-tenant Based on SaaS. Dalian Maritime University, China.
- Wei, Y., 2009. Software Revolution of SaaS Architecture Design in the Internet Age. Electronic Industry Press, Beijing.