Chapter 77
Design and Implement of Information System for Water Management Based on SaaS

Lilong Jiang, Junjie Chen, Zhijie Lu and Junjie Li

Abstract According to the problems of the high cost in the development and maintenance of the information system for water management, the information system for water management based on Software as a Service was designed and implemented. With the information system, the Water Supplies Department can make customary configuration, sell water and do data analysis and so on. This paper describes the design and implement of SaaS in the information system for water management. It can serve to reduce the operating cost of the Water Supplies Department and finally promote the informatization of the water industry.

Keywords SaaS · Configurability · Information system for water management · XML

77.1 Introduction

Recently, with the development of information technology, the information system for water management is more and more applied into the water industry, which improves the efficiency of water management greatly [1]. But the traditional information system for water management is for some water supplies department, the cost of purchasing, maintaining and updating software is relatively high. What's more, the technical workers must be employed by the water supplies department to maintain the software. Software as a Service (SaaS) arises as a new software application mode with the development of internet and the mature of

L. Jiang (✉) · J. Chen · Z. Lu · J. Li
Northeastern University, Complex Building 118, Shenyang, Liaoning, China
e-mail: jianglilong0225@gmail.com

X. He et al. (eds.), *Computer, Informatics, Cybernetics and Applications*, Lecture Notes in Electrical Engineering 107, DOI: 10.1007/978-94-007-1839-5_77,

Table 77.1 SaaS maturity model

	Configurability	Multi-tenant-efficiency	Scalability
Level 1	×	×	×
Level 2	√	×	×
Level 3	√	√	×
Level 4	√	√	√

application software [2, 3]. In the SaaS model, the users don't have to purchase software and hardware, construct the IT room and employ technical workers any longer while they can make use of the system by online software rental. This system is an information system for water management based on SaaS. The water supplies department can use the system through internet and order the necessary software services by actual demand. As a result, the running cost of the water supplies department is reduced greatly.

77.2 Introduction of SaaS

SaaS is a new commercial model which software is deployed as managed service and provides service through the internet. The client accesses to services by means of rental. And according to their actual needs they can order relative services which is charged by the content and the duration of the service. The service provider provides upgradable, multi-tenant and configurable services, which simplifies the deployment and maintenance of software. According to configurability, multi-tenant-efficiency and scalability, the SaaS applications maturity can be expressed using a model with four distinct levels, which is given in Table 77.1.

Level 1 and Level 2 Maturity Model can cause high cost of hardware and running, for they design a separate instance for each tenant. Although in terms of architecture, Level 4 Maturity Model with configurability, multi-tenant-efficiency and scalability is the most ideal application architecture, this level needs Load Balance Level and multi-application server and it also involves the complex problems of data horizontal split, a high demand for hardware and software and so on. So the Level 3 Maturity Model is the real SaaS application basic architecture and it is more suitable for the application of Information System for Water Management.

77.3 Design and Implement of Level 3 Maturity Model

77.3.1 Multi-Tenant Data Architecture

There are three approaches to manage Multi-Tenant Data [4, 5], which are "Separate Databases", "Shared Database, Separate Schemas" and "Shared Database, Shared Schema". The comparison of all three approaches is showed in Table 77.2.

Table 77.2 Comparison of three approaches to the management of Multi-Tenant Data

Approach	Isolation level	Share level	Security	Cost
Separate database	High	Low	High	High
Shared database Separate schemas	Middle	Middle	Middle	Middle
Shared database Shared schema	Low	High	Low	Low

In order to reduce the cost of hardware and maintanance, the information system for water management will adopt the approach of "Shared database, Shared schema". And this will make the system provide service for more customers with less server resources.

77.3.2 Custom Data Model

Custom Data Model is made up of entitiy-predefined meta data, entity instance's data and tenant's extended meta data. Meta data consits of DataNodes table and DataProperties table, which both of them constitute the data dictionary. DataNodes connecting with each other makes up a tree by parentId and DataProperties table describes the properties of DataNode. The definitions of entity and field both correspond the DataNodes. NodeType is used to distinguish Entity whose value of nodetype is 1 from Field whose value of nodetype is 2. Every row in DataNodes has a TenantId. DataNodes whose tenantId is NULL represent the system predefined meta data while others mean the users' extended meta datas. It is the same in DataProperties. The relationship digram of database is showed in Fig. 77.1.

Taking into the consideration that different tenant has different price model, in the information system for water management; the price model was created. The price type entity consists of two sub DataNodes which are price DataNode representing the price of the price type and price type representing the type of price. Tenants can extend the predefined price type entity, for example adding pollution price DataNode.

77.3.3 Configurable Invoice

Tenants in the system need to print invoice. But the invoice format differs in different tenants and the common Crystal Report can't print precisely. The system describes the invoice format with XML, therefore the precise printing and configurable invoice were solved. The XML definition is showed in Fig. 77.2.

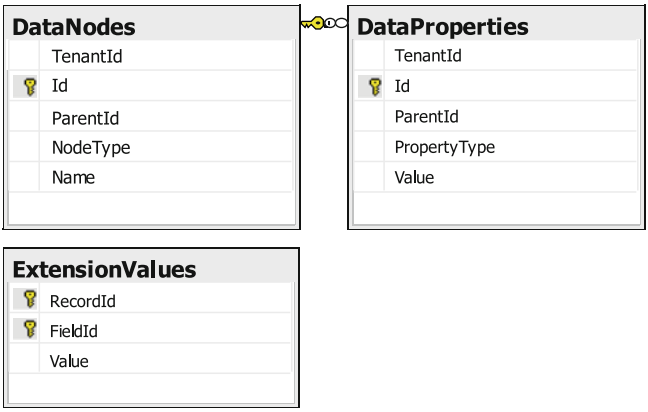

Fig. 77.1 Relationship diagram of database

```
<root>
    <pagesetting>
        <Landscape>false</Landscape>
        <paperkind>A4</paperkind>
        <paperwidth>200</paperwidth>
        <paperheight>100</paperheight>
        <pageleft>10</pageleft>
        <pageright>10</pageright>
        <pagetop>10</pagetop>
        <pagebottom>10</pagebottom>
    </pagesetting>
    <reporttable>
        <text field="CName" value="Customer Name" x="20" y="20"
        fontsize="10" />
        <text field="NULL" value="Org Name" x="120" y="20"
        fontsize="10">Hefei Water Company </text>
        <text field="Price" value="Price" x="20" y="70" fontsize="10" />
        ……
    </reporttable>
</root>
```

Fig. 77.2 XML definition for invoice template

Pagesettings configuration section defines the page settings of the invoice while reporttable configuration section defines the printing items of the invoice. The printing item of the invoice is defined by Text configuration section. The field attribute of Text is responsible for the binding database field while value attribute marks the item and points out the meaning of the item. If field is NULL, it means this item's content is fixed and it doesn't need to fill up. X and Y attributes are the coordinate used to locate the printing content. Fontsize defines the font size of the content.

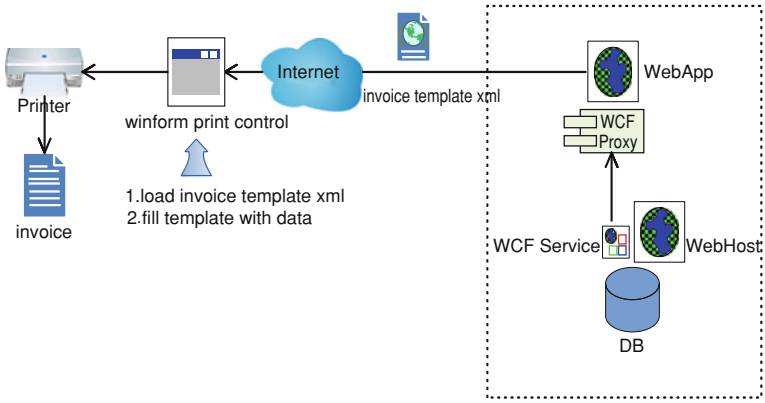

Fig. 77.3 Process of printing invoice

Figure 77.3 illustrates the process of invoice printing. The tenants customize their own invoice template and then the template is saved in the database. When printing invoice, the invoice template will be taken out of the database. On clicking the printing button, the water-selling data will fill up the template and the template will be sent to the winform printing control embedded in the asp.net page which aims to parse the XML and accomplish the task of printing invoice.

77.3.4 Configurable User Interface

The configurable user interface is made up of two parts which are the configurable system menu and the configurable page elements.

77.3.4.1 Configurable System Menu

In system, each tenant owes one menu and each sub menu item corresponds one atomic function. The class diagram of class MenuLink and its corresponding parse class MenuParse in showed in Fig. 77.4.

The menu information of each tenant is stored in XML. The XML definition is showed in Fig. 77.5.

77.3.4.2 Configurable Page Elements

As the same with the system menu [6], the content on every function page is also responsible for the interaction between the user and the system. So different tenant will have different demands, whatever the demand are the number, location and showing sequence of the page elements.

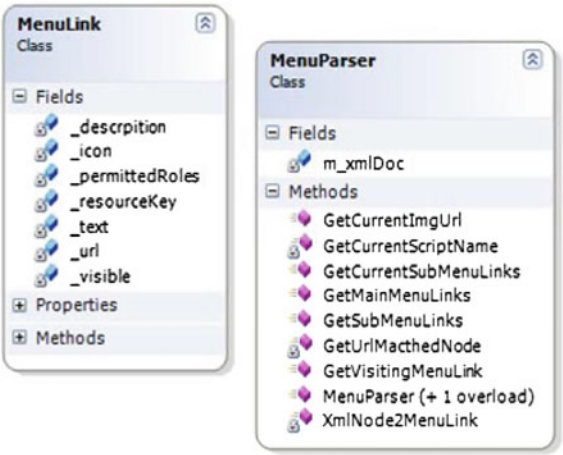

Fig. 77.4 The class diagrams of MenuLink and MenuParser

```
<ArrayOfMainMenu xmlns="">
    <MainMenu Text="Water Sale" Url="~/SaleWater/WaterSaling.aspx"
    Roles="Administrator,Salesman" Icon="~/Images/task.gif" >
        <SubMenus>
          <SubMenu Text="Sell Water" Url="~/SaleWater/WaterSaling.aspx"
          Icon="~/Images/admin.gif" Roles="Administrator" Depends="Print
          Invoice"/>
          <SubMenu Text="Print Report" Url="~/Rpt/DateSale.aspx"
          Icon="~/Images/print.gif" Roles="Administrator" />
          <SubMenu Text="Print Invoice" Url="~/SaleWater/InvoicePrint.aspx"
          Icon="~/Images/print.gif" Roles="Administrator" Visible="false"/>
        </SubMenus>
    </MainMenu>
     ...
</ArrayOfMainMenu >
```

Fig. 77.5 XML definition for tenant's menu

The implement of configurable page elements is mainly through adopting the technologies of Themes and WebPart in .NET. Tenants can replace their themes or skins, add, modify and delete their own webparts.

77.3.5 Configurable Functions

SaaS stresses "use on demand" and "pay per use" [7]. This is a good solution for the differences in the different tenants' demands and different demands in different times for the same tenant. The function packages in the system are showed in Fig. 77.6.

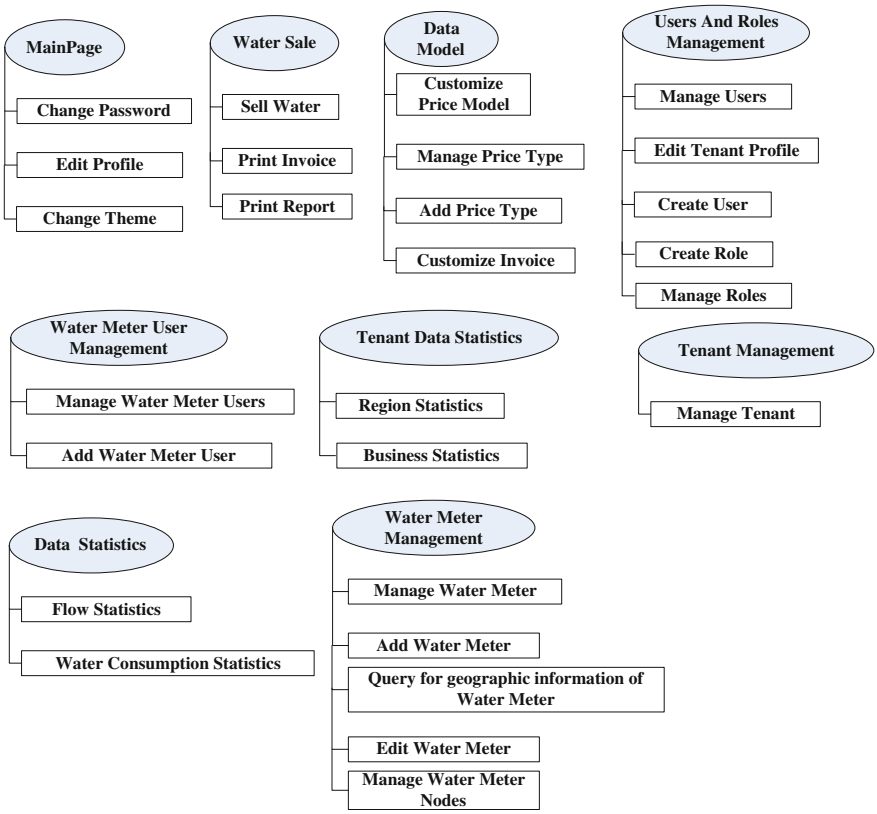

Fig. 77.6 Function Packages

The corresponding relation between function package and menus is stored in XML file showed in Fig. 77.5. Each MainMenu label corresponds one function package and each SubMenu label corresponds one atomic function. The dependence relation between atomic functions is embodies by the Depends attribute in SubMenu. Since each tenant has his own XML file, the configurable functions are implemented. What's more, through Roles attribute in MainMenu and SubMenu, Tenants can create different roles with different rights.

77.4 Conclusions

This chapter introduces basic contents about SaaS, designs and implements the information system for water management based on SaaS. This system can serve to reduce the cost of the water supplies department's purchasing and maintaining

software. Moreover it can enhance the construction of information. Research on SaaS can be referenced by other similar systems.

References

1. Ye W (2009) The software revolution in internet times-design of SaaS archiecture. PHEI, Beijing (in Chinese)
2. Meng Z (2007) Application development of SQL server 2005. TsingHua University, Beijing (in Chinese)
3. Software+Services. http://msdn.microsoft.com/en-us/library/bb977472.aspx
4. Wen N (2010) The research of GSP medicine management system s security based on SaaS. CNKI:CDMD:2.2010.166049 (in Chinese)
5. Zhao B (2009) The application research of SaaS maturity model level 3 in buy-out model. CNKI:CDMD:2.2009.167654 (in Chinese)
6. Mietzner R, Leymann F, Papazoglou MP (2008) Defining composite configurable SaaS application packages using SCA, variability descriptors and multi-tenancy patterns. 2008 3rd international conference on internet and web applications and services, pp 156–161
7. Byanjankar S, Winoto P, Paik W (2010) In: Proceedings of the 2010 sixth international conference on networked computing and advanced information management (NCM 2010), pp 440–445