International Journal of Future Generation Communication and Networking
Vol. 10, No. 1 (2017), pp. 159-170
http://dx.doi.org/10.14257/ijfgcn.2017.10.1.14

Design and Implementation of a Lightweight Electronic Village System Based on REST Web Service

Binfeng Li [1], Jianguo Chen [1, *], Changyu Liu [1] and Yilong Li [1]

[1] *College of Mathematics and Informatics, South China Agricultural University, Guangzhou 510642, China*
*binfeng.lee@gmail.com, *cjg63@qq.com (corresponding author), yezhich@gmail.com, 1217066840@qq.com*

Abstract

Electronic village system is becoming increasingly essential because of the significance of helping rural informatization and narrowing digital divide between rural area and urban area. *However, traditional electronic village systems cannot meet the demand of rural area nowadays. This paper proposes a well flexible, scalable and maintainable lightweight electronic village system that is adequate for practical use based on REST web service. The proposed system not only satisfies the functional requirements of publics in rural area, but also has good architectural characteristics and adaptability to rapid changing informatization demand, compared with other electronic village systems. Moreover, the proposed system is able to allow systems of higher-level government to access service by CORS, and it supports multiple types of clients, including PC and mobiles. The electronic village system will come into use in Minxing Village located in Guangzhou City soon.*

Keywords*: REST web service, electronic village system, resource oriented architecture*

1. Introduction

Electronic village systems (E-village systems) have been attracting more and more attention from government and publics in rural area [1], since it can not only provide online public affairs service and information for villagers, but also contribute to rural informatization. Moreover, electronic village systems can help to change the long-term dilemma of digital divide between rural and urban area in China [2].

In recent years, researchers have begun to dive into the study of electronic village systems. Most of them pay attention to the requirements and functionalities of electronic village systems. The functionalities of electronic village systems mainly include managing information of village committee [3-4], releasing public information [5] and offering public affairs service for villagers [6-8] in rural area. However, the non-functional factors of electronic village system are not heeded enough, so that it caused some problems in practical, especially the problems relevant to architecture. These traditional electronic village systems cannot follow the rapid change of rural requirements and satisfy the fast growth of rural informatization. Besides, the traditional electronic village systems are isolated and enclosed, which makes it hard to integrate resource and share information between different systems. A few researchers focus on the architecture or framework of electronic village systems in order to build more open and flexible systems. Yang *et al*. propose a high-level architecture called CC-REGA based on cloud computing [9]. Huang *et al*. propose a framework for rural information resource integration combined with the characteristics of Chinese rural area [10]. The architectures they proposed are more reasonable and adaptable in current environment. However, these architectures are so complicated and heavy that it is hard to put them into practical

ISSN: 2233-7857 IJFGCN
Copyright ⓒ 2017 SERSC

use at present, considering the high cost of construction and operation, poorer infrastructure than urban area and deficiency of IT technicians in rural area. Therefore, it is imperative to design a lightweight electronic village system that is more suitable for rural area in China.

For the problems of previous electronic village systems, this paper dedicates to design and develop a lightweight electronic village system based on REST web service. The system has following characteristics: (1) The proposed system is more lightweight and simple than previous systems, using a lightweight framework to build REST web service that is lightweight and simple. (2) The system has better flexibility, scalability and maintainability benefiting from REST web service than traditional SOAP-based web service, especially in distributed environment. (3) The system is hypermedia-driven that meets HATEOAS constraint, which decouple the client and server. Thus, client and server can independently evolve. (4) The system allows systems of higher-level government to access service for acquiring wanted data through CORS (Cross-Origin Resource sharing). (5) The system can support multiple types of clients, including PC and smart phone.

2. Requirement Elicitation

2.1. Overview to Functional Requirements

Functional requirements depict the interactions between the developed system and its environment independent of its implementation. The environment includes the user and external systems with which the developed system interacts. The main activities of functional requirements elicitation are identifying actors, scenarios, use cases, relationship among actors and use cases.

In the proposed electronic village system, there are three actors, which are villagers, village committee, and systems of higher-level government, such as the system belong to township government. Villagers are core users who can gain public affair services and public information from village committee through the system. Village committee is an abstraction user because village committee is composed of staffs who work in village committee in reality. The staffs in village committee use the system to fulfill their tasks and serve villagers, for example, managing villages' information, releasing various public information and collect statistical data. The system of higher-level government like township government can access the service from electronic village system so that township government can acquire wanted data, especially those statistical data. For each actor, some use cases can be abstracted from the daily scenarios they involve. The above actors engage in the use cases as illustrated in use-case model shown in Figure 1.

 Copyright ⓒ 2017 SERSC

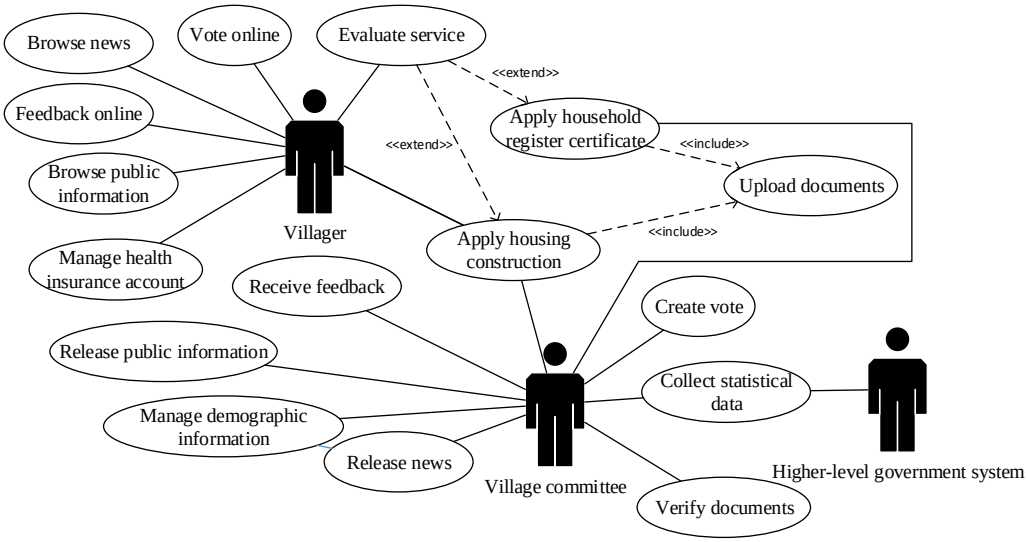

Figure 1. Use-case Model

2.2. Overview to Nonfunctional Requirements

Nonfunctional requirements describe aspects of developed system that are not directly related to functional behavior of the system. Robert Grady proposed the FURPS+ model used by Unified Process to capture nonfunctional requirements [11]. FURPS is an acronym of five categories requirements: functionality, usability, reliability, performance and supportability. The "+" in FURPS+ represents identification of additional requirements subcategories that represent constraints, which include design constraint, implementation constraint, interface constraint and physical constraint.

Functionality: For the security, users have to be authenticated. The staffs in different departments of village committee have different authorities.

Usability: User interface is in simple style and the system is easy to use since many elderly villagers are not skillful in computer. System should display meaningful error message to remind uses how to correct their inputs.

Reliability: System should be normally running at weekday as possible.

Performance: The system should support at least 100 concurrent users. Villagers cannot feel a sense of latency when the system is not busy.

Supportability: Villagers can use system without any special software installation. System should provide log for the convenience of maintenance.

Implementation constraint: Open source technologies should be used to develop system considering the ease of maintenance and cost-saving.

3. Design of Lightweight Electronic Village System

3.1. Resource Oriented Architecture Design

Resource Oriented Architecture (ROA) is based on the concept of resource [12]. Resource in the system is directly addressable distributed component that is handled by a standard and uniform interface. Platforms based upon REST enable the creation of resource oriented architecture. So architectures that satisfy the constraint of resource oriented architecture are considered to conform to REST. REST is an architecture style that is independent from any protocol such as HTTP or implementation mechanism. Therefore, this paper uses resource oriented architecture when talking about the concrete system architecture.

The architecture of lightweight electronic village system is shown in Figure 2. In terms of functionalities and needs, electronic village system is composed of several service

components. Every service component is partitioned into four layers, including controller layer, representation layer, business logic layer and data access layer. Resource layer of system is shared by all service components. Clients can access a service through the URI and handle a resource that the URI points to through common HTTP interface.

3.2. REST Web Service Design

Generally, the design of REST web service follows several steps: figuring out the data set, designing resource and interface, selecting resource representation and describing REST web service workflow.

REST web service is based on resource, and resource has close relationship with data set. According to the requirements, data set of the system can be figured out. The entity and relationship elements in E-R model are used here to denote the data set and the relationship between data as shown in Figure 3. After the data set is decided, the resources of system can be identified based on data set. Then URI and interface (HTTP verbs) should be designed for every resource as shown in Table 1.

Representation of resource is a hypermedia format that denotes the transformation form of resource state at a certain time. Representation of resource can be encoding in various format, such as XHTML, JSON and Atom. JSON is one of the most popular lightweight data format in web and mobile application. Considering that, the service of electronic village system is shared for web client and mobile client, and JSON is better supported in web client using AJAX, so JSON is the best choice for the system.

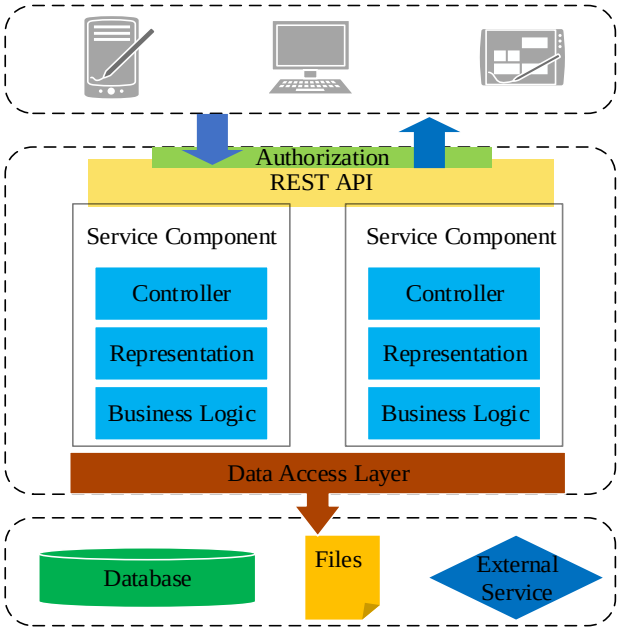

Figure 2. Architecture of Lightweight Electronic Village System

 Copyright ⓒ 2017 SERSC

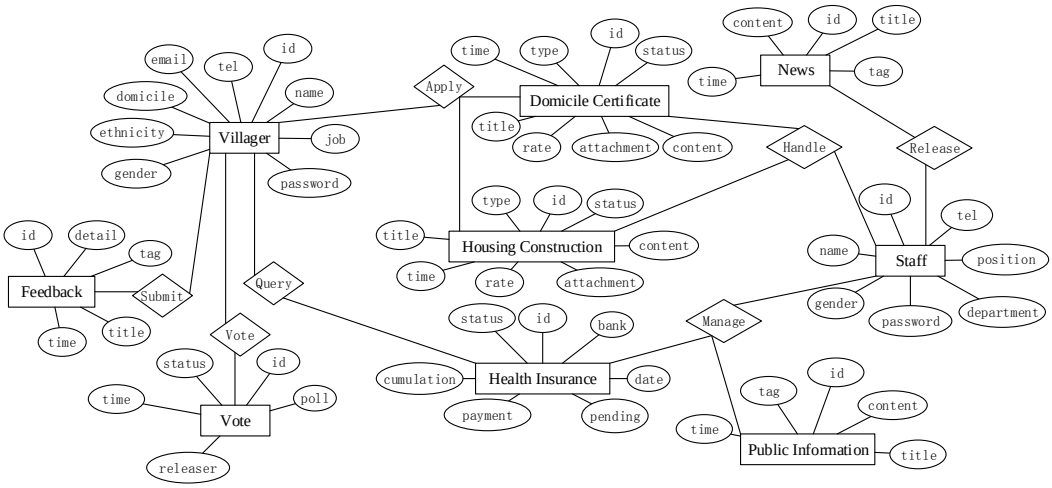

Figure 3. Dataset of Lightweight Electronic Village System

Table 1. URI and Interface Design for Resources

Resource	URI	Interface
Villager	/villagers/{Id}	GET POST PUT DELETE
Staff	/staff/{Id}	GET POST PUT DELETE
Domicile certificate	/certificates/{villager}/{Id}	GET POST PUT DELETE
Housing construction	/constructions/{villager}/{Id}	GET POST PUT DELETE
Health insurance	/insurances/{villager}/{Id}	GET POST PUT DELETE
Public information	/information/{Id}	GET POST PUT DELETE
News	/news/{Id}	GET POST PUT DELETE
Feedback	/feedback/{villager}/{Id}	GET POST
Vote	/votes/{Id}	GET POST
document	/documents/{villager}/{Id}	GET POST PUT DELETE
Statistics	/statistics/{Id}	GET POST

REST web service workflow regards a state of a step in the workflow as a state of resource. The state of resource transfers through request and response of REST web service. In this paper, a concept of extension finite state machine is introduced to describe REST web service workflow. The definition of extension finite state machine is a quintuple

$$(\sum, S, S_0, \delta, F)$$

where:

$\sum$ is the input alphabet that is finite and non-empty set of symbols. Each element in the set is triple denoted by$\{i, u, c\}$. In this tri-tuple, $i$is a HTTP uniform interface, and$u$ is URI for resource and cis HTTP response code.

S is the set of states that is finite and non-empty. $S \neq \Phi$.

S_0 is the initial state. $S_0 \in S$.

δ is the state transition function. $\delta: S \times \sum \rightarrow S$.

F is the final state. $F \in S$.

Handling domicile certificate is taken as an example to elaborate the REST web service workflow with finite state machine as shown in Figure 4. Villager can apply for domicile certificate in the system by sending HTTP POST request from client, and system then creates a new resource of domicile certificate. Villager can cancel or upload required documents by sending HTTP DELETE or HTTP POST request. After that, staff in village committee receives the application of villagers and then verifies documents uploaded by villager. If the documents of villager are verified successfully, staff fills in and confirms information of domicile certificate and system updates information of domicile certificate

by HTTP PUT. If the documents of villager is verified failed, staff can ask villager for re-uploading documents or cancel the application of villager. Formal description based on finite state machine of handling domicile certificate workflow is as follow.

$\sum = \{apply, upload, reupload, cancel, verify, confirm\}$, where:

apply $= \{POST, /certificates/\{villager\}, 201\}$;

upload $= \{POST, /documents/\{villager\}, 201\}$;

reload $= \{PUT, /documents/\{villager\}/\{Id\}, 200\}$;

cancel $= \{DELETE, /certificates/\{villager\}/\{Id\}, 200\}$;

verify $= \{PUT, /certificates/\{villager\}/\{Id, 200\}$;

confirm $= \{PUT, /certificates/\{villager\}/\{Id\}, 200\}$;

$S = \{S_0, S_1 = Created, S_2 = Uploaded, S_3 = Canceled, S_4 = Verified, F = Confirmed\}$

The transition functions δ are:

$(S_0, apply) = S_1, (S_1, upload) = S_2, (S_1, cencel) = S_3,$
$(S_2, verify) = S_4, (S_2, reupload) = S_2, (S_2, cancel) = S_3,$
$(S_3, confirm) = F.$

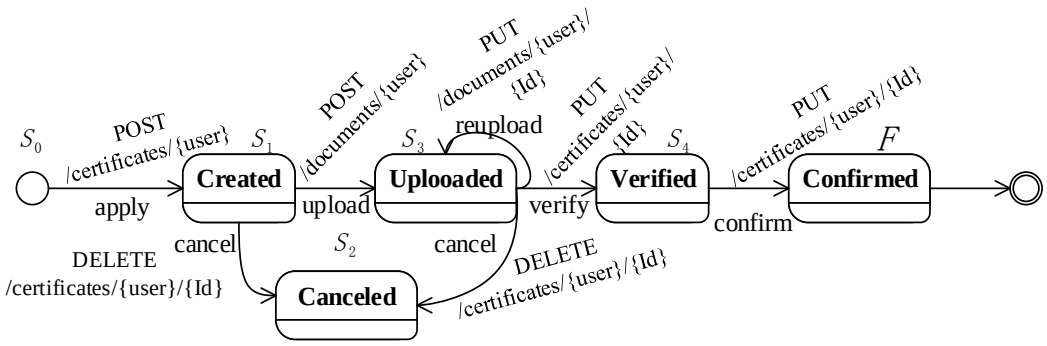

Figure 4. Finite State Machine of Handling Domicile Certificate

4. Implementation of Lightweight Electronic Village System

4.1. Implementation of REST Web Service

For lightweight electronic village system, Spring MVC and Spring HATEOAS is used to build REST web service. Spring HATEOAS provides APIs to ease creating REST representations that follow the HATEOAS principle when working with Spring MVC.

The flow of building REST web service is shown in Figure 5. The request sent from client is intercepted to validate the user's authorities. Then Controller invokes the method annotated with the same URI and HTTP verb in the request using @RequestMapping. The method will handle resource that client request according to HTTP verb, such as creating a new resource using HTTP POST. After the resource is handled, system creates resource links and relation between links to guide the client that next potential application state (available operation on a resource). This step is important because it is the constraint to distinguish real REST web service. Then system will wrap the resource (object) that client requests and links into JSON that is sent to client.

The key classes and interfaces involved in the above REST web service flow is shown in Figure 6. RestController maps the URI to a method that handle the request and response to client with resource encapsulated in JSON. ResourceAssemblerSupport base class dedicates to map an entity to a resource type. Assembler class that extends ResourceAssemblerSupport can either override toResource method that assemble a single resource or override to toResources method that assemble a set of resources. ControllerLinkBuilder creates nested links by pointing to controller classes, so it can help

 Copyright ⓒ 2017 SERSC

create flexible links, which can avoid hard coding of links. Generic service design pattern is applied to business logic layer, in order to reduce tedious code. Business logic layer is in charge of transaction management besides business handling. JpaRepository is the interface in data access layer provided by Spring JPA to ease interacting with database.

According to the requirements that the system should provide higher-level government's systems with the permission to acquire statistical data and support smart phone client, so the system have to support CORS (Cross-Origin Resource Sharing). Spring framework offer APIs to ease the implementation of CORS. The configuration parameters of CORS in Spring is shown in Table 2.

Table 2. Configuration of CORS

Parameters	Description
allowed-origins	Set the origins to allow to access.
allowed-methods	Set HTTP methods that is allowed. For villagers' client, POST and GET method is allowed. For systems of higher-level government, GET method is allowed.
allowed-headers	Set the list of headers that a pre-flight request can list as allowed for use during an actual request.
allowed-credentials	Whether user credentials are supported.
max-age	Configure how long, in seconds, the response from a pre-flight request can be cached by clients.

4.2. Implementation of Web Client

AngularJS is applied to build Ajax web client of lightweight electronic village system. AngularJS provides built-in service called $http to interact with REST web service API on server. $http service support common HTTP methods for Ajax request as shown in Table 3. AngularJS uses JavaScript promise to handle Ajax response. Table 4 shows the methods defined in promise returned by the methods shown in Table 3. When server successfully responses and returns JSON to client, AngularJS parses JSON and create JavaScript objects. Then the JavaScript objects are passed in success function so that they can be handled and presented.

Copyright ⓒ 2017 SERSC

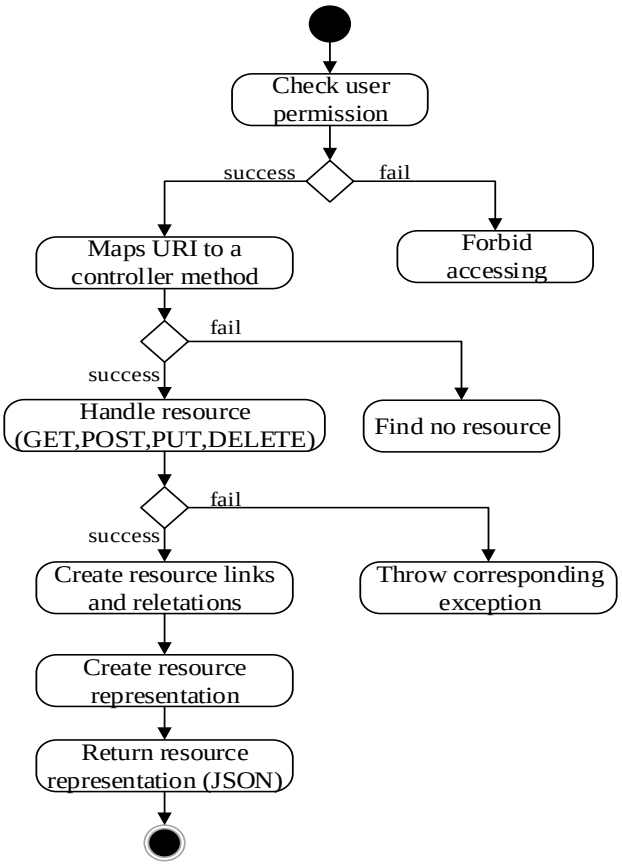

Figure 5. Flow of Building REST Web Service

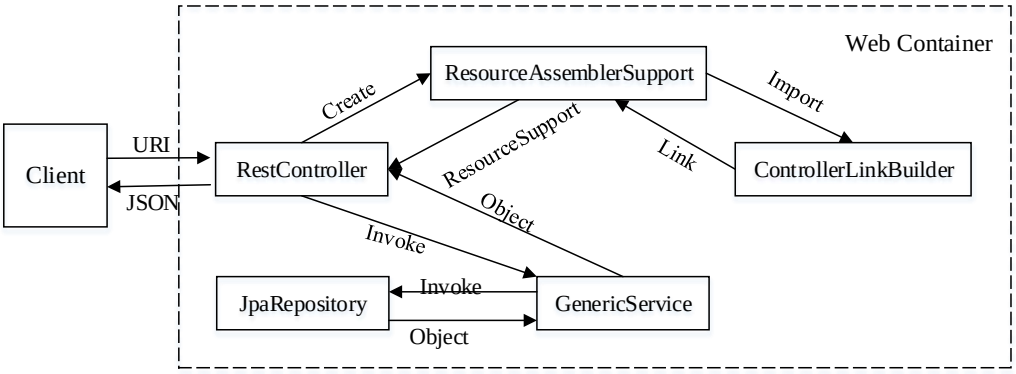

Figure 6. Key Class and Interface of REST Web Service

Table 3. Methods Defined in $http for Ajax Request

Methods	Description
get(url, config)	Send GET request to given url.
post(url, data, config)	Send POST request to given url with data.
put(url, data, config)	Send PUT request to given url with data.
delete(url, config)	Send DELETE request to given url.
head(url, config)	Send HEAD request to given url.

 Copyright ⓒ 2017 SERSC

Table 4. Methods Defined in Promise

Methods	Description
success(fn)	Invoke specified function when request is successful.
error(fn)	Invoke specified function when request is failed.
then(fn)	Register success function and failure function.

Bootstrap is used for developing aesthetic and responsive layout web pages faster and easier. So lightweight electronic village system can adapt to various devices with different screen sizes. The home page of system is shown in Figure 7.

Figure 7. Homepage of Lightweight Electronic Village System

5. Conclusion

As electronic village system gains attention in rural China, it becomes important to study how to design and develop an electronic village system that is adequate for practical use and adapt to rapid changing requirements along with fast growth informatization in rural area. In this paper, a lightweight electronic village system based on REST web service that has good characteristics, such as scalability, flexibility and simplicity, is designed and implemented to serve for rural areas. The lightweight electronic village system presents a new idea for the construction of electronic village system and other systems in rural area. The proposed system soon will come into use in Minxing Village located in Guangzhou, China. In the near future, our research will focus on the REST web service composition and mobile client development for improving the system.

Acknowledgments

This paper was supported by the Science and Technology Planning Project of Guangdong Province, China under Grant No. 2016A020210103, and the Scientific and Technological Innovation Project of University Students in Guangdong Province under Grant No. 201510564286. We would like to thank anonymous reviewers for helpful comments.

Copyright ⓒ 2017 SERSC

References

[1] X. Kong and J. Zou, "Review on Status Quo and mode of rural E-government in China", E-Government, no. 1, **(2015)**, pp. 90-96.
[2] L. Zhao, G. Liao and Y. Chen, "Status Quo and Development Countermeasures of Village-Level E-government", Agricultural Engineering, vol. 2, no. 9, **(2012)**, pp. 23-26.
[3] L. Xia, "Studies on the Role and Construction of Rural E-Government", Hubei Agricultural Sciences, vol. 52, no. 21, **(2013)**, pp. 5354-5357.
[4] H. Wang, "On the Construction of Electronic Village Administration in New Countryside", Journal of Jiangxi Agricultural University, vol. 8, no. 2, **(2009)**, pp. 72-75.
[5] J. Cao, K. Zhu and Y. Fan, "Design and Implementation of Information Service Management System in Rural Area", Jiangsu Agricultural Sciences, vol. 40, no. 5, **(2012)**, pp. 380-382.
[6] F. Wang, J. Zheng and L. Wang, "Design and Development of E-Village Affairs System", Shangdong Agricultural Sciences, vol. 46, no. 5, **(2014)**, pp. 18-20.
[7] R. Xu, D. Jia and X. Zhou, "Study on the Construction of Rural informatization platform in China", Journal of Nanjing University of Posts and Telecommunications (Social Science), vol. 15, no. 4, **(2013)**, pp. 14-20.
[8] M. Ma, X. Qin and J. Li, "Status of rural E-government and research of its system construction", Agricultural network information, no. 11, **(2008)**, pp. 96-99.
[9] Q. Yang, M. Huang and B. Wan, "A Rural E-Government Architecture Based on Cloud Computing", Computer & Digital Engineering, vol. 39, no. 10, **(2011)**, pp. 25-29.
[10] L. Huang, Q. Luo and C. Zou, "A Framework of New Rural E-Government and the Related Information Resources Integration", International Journal of u- and e-Service, Science and Technology, vol. 6, no. 6, **(2013)**, pp. 83-96.
[11] L. Zong, "Models of Management of Software Quality", Journal of Hubei University of Education, vol. 28, no. 2, **(2011)**, pp. 76-78.
[12] C. Pautasso, "RESTful Web service composition with BPEL for REST", Data & Knowledge Engineering, vol. 68, no. 9, **(2009)**, pp. 851-866.

Authors

Binfeng Li, he is a graduate student in College of Mathematics and Informatics at South China Agricultural University. He received his bachelor's degree in 2013 from South China Agricultural University. He was secretary of Graduate Student Youth League Committee in College of Mathematics and Informatics at South China Agricultural University from May 2014 to June 2015. His research interests mainly include management information system and decision support system of agriculture.

Jianguo Chen, he received the PhD degree in Management Science and Engineering fromCentral South University in 2007. He is an professor at the College of Mathematics and Informatics, South China Agricultural University. He is the head of the discipline of Management Science and Engineering&Industrial Engineering at the College of Mathematics and Informatics. He is a member of the Academic Committee of South China Agricultural University. His research interests include industrial engineering and enterprise informatization.

Changyu Liu, he received the PhD degree in 2015 from South China University of Technology, where he worked under the supervision of Prof. Shoubin Dong. He is currently a lecturer at the College of Mathematics and Informatics, South China Agricultural University. He was a visiting scholar at the School of Computer Science, Carnegie Mellon University, from September 2012 to October 2013, advised by Dr. Alex Hauptmann. Then, he worked with Prof. Mohamed Abdel-Mottaleb and Prof. Mei-Ling Shyu at the

 Copyright ⓒ 2017 SERSC

Department of Electrical and Computer Engineering, University of Miami, from October 2013 to September 2014. He serves as a reviewer for many international journals, such as Neural Computing and Applications, Security and Communication Networks, KSII Transactions on Internet and Information Systems, Journal of Computer Networks and Communications, and Tumor Biology. He is a Technical Program Committee member for many international conferences, such as GMEE2015, PEEM2016, ICEMIE2016, and WCNE2016. His research interests include computer vision, pattern recognition, machine learning, bioinformatics, virtual reality, and multimedia analysis.

Yilong Li, he is an undergraduate student who majors in Industrial Engineering at College of Mathematics and Informatics in South China Agricultural University. He is in charge of the Scientific and Technological Innovation Project of University Students in Guangdong Province under title “Key Techniques Study of Electronic Village Mobile Cloud Platform” now. And he is the monitor of his class. His research interests include enterprise operation management and supply chain management.

Copyright ⓒ 2017 SERSC

Copyright ⓒ 2017 SERSC