International Journal of u- and e- Service, Science and Technology
Vol.8, No. 11 (2015), pp.389-402
http://dx.doi.org/10.14257/ijunesst.2015.8.11.37

A Customized Web Service Constructing Model Based On Private Cloud

Weng Yu and Peng Guo

College of Information Engineering, Minzu University of China
dr_wengyu@126.com

Abstract

Varieties of web services are available in the internet, but we always be the users even we have submitted our data to a cloud platform. Then how to let a user who has no programming basics be a service provider using his or her own data? In this paper, we propose a web service constructing model which can help users construct their own web service. And the experiment reveals that all kinds of the data source provided by users in private cloud is demonstrated properly through our cloud platform using Web GIS. In other words, users can construct web service with their own data through simple configuration.

Keywords*: web service; WSDL; service constructing; private cloud; cloud computing*

1. Introduction

Among the current Internet, there is a wide variety of private cloud storage services which are available to users. They support considerable storage space, but only a single function of storage [1]. The data do not allow users to further processing. The right to deal with these data only rests in the hands of the storage service providers.

How to make individual users construct their own web services [2] with their own data is an initial question needs to be solved. What's worse, if the user does not have any programming foundation, the question will be more difficult in this environment. The ideal solution is few text input and simple mouse clicks. To simplify user operations and shield the programming details, in this paper, we propose a model for individual users to construct their own services. Data sources of this model come from the user stored in the user's private cloud [3], and the final data results will take web pages using Web GIS for example.

The rest of the paper is organized as follows: Section 2 describes the related works about web service constructing and web service description language. Section 3 gives the description of the problem with more details. Then the components and data flows of the model are talked about in Section 4. And Section 5 is the procedure of experiment. Then the conclusion is made in Section 6 and we talk about some future plans in Section 7.

2. Related Works

2.1. Web Service Description Language[4]

WSDL, as a widely used language, is playing an important role in web service publishing, retrieval, *etc*. And it is also a significant part in the model we proposed. Some people have done lots of work to enhance the scalability and practicality. Parimala N and Anu Saini put forward X-WSDL, they bring a new element named 'criteriaservice'[5] into WSDL. And the element is available in the new namespace. Additionally, using this extension allows to specify the criteria along with the service in X-WSDL document. In addition, six steps were proposed by Marco Crasso and his colleagues to revise WSDL

ISSN: 2005-4246 IJUNESST

documents, especially to avoid the anti-patterns when we use the contract-first approach[6]. Nevertheless, code-first is the more popular way we use than contract-first. So they give other ways to avoid anti-patterns for code-first in another paper [7]. Banage T. G. S. Kumara and his colleagues extract features from WSDL documents to help understand web services[8]. Learning from their points of view, we cited another simple XML description file through “import”, in addition to completing the conventional elements of description file. It helps to improve the componentization, completeness of the WSDL, because this file contains the implementation details about the internal services. A more detailed document structure will specifically addressed in the experimental section that follows.

2.2. Web Service Publishing

Web service publishing is an important step of a web service’s life cycle[9], and a service description file is a key point to improve the efficiency of service constructing and follow-up release. In other ways, A Service Broker Based Architecture for Effective Discovery and Automatic Publishing of Internet Web Services was proposed by Demian Antony D'Mello and his colleagues to improve the efficiency of discovery and auto-publishing[10]. Among their paper, they presented through a series of activities to complete automatic web service publishing: firstly, WSDL crawler would guide to the location of a service description file. Secondly, the WSDL file will be downloaded and stored to WSDL repository by WSDL down loader. Then these files would be passed to a matchmaking module. Then, calculate the level of each respective service during constantly updating process according to the data type and WSDL file. Lastly, complete the service publishing work in accordance with the WSDL information and the level which has been calculated. This approach is also for enterprise or organization, made no specific guidance for individual user. In this paper and at this stage, we firstly aim to help the individual users who have own data to construct a web service and then allows people to get in touch with this service.

3. Description of the Problem

Nowadays, data as an extremely important network resource, its utilization did not meet the people's expectation. Cloud computing[11, 12, 13, 14] technology to flourish, greatly enhanced the ability of data processing. But in reality we often encountered such problems: some organizations have powerful data processing tools, but there is not enough data sources for them to achieve the desired result; to the contrary, some organizations even some individuals hold a good deal of data resources with all kinds of data storage media. At the same time, the content is great variety, which can not even find the appropriate treatment; Indeed, there are a considerable part of the organizations do not even know the value of their data in the hands. Their purpose is to protect them rather than amplify the value of the data. Most of the people there have no computer programming background and foundation. In our model, such people are our important target customers. Under normal circumstances, if they want to process the data for some reason, they accustomed to using services provided by IT companies. In this case, the direct operation of user’s data will also be handed to others, which is prejudicial to the security of user data and their private cloud.

Two important issues of the question: no programming basics and construct the web service by themselves using their own data.

We want to provide the user with a simple private cloud space, then let the user construct his or her own web service just through simple interface which we only reserved for the user the appropriate buttons and input boxes. We shield back the implementation details as far as possible. We only check the results of the user's service constructed. As we only provide a pot, what type of raw food will be used is the user's freedom, and we just check the last of the dishes is non-toxic and can be eaten.

4. Components and Data Flows of the Model

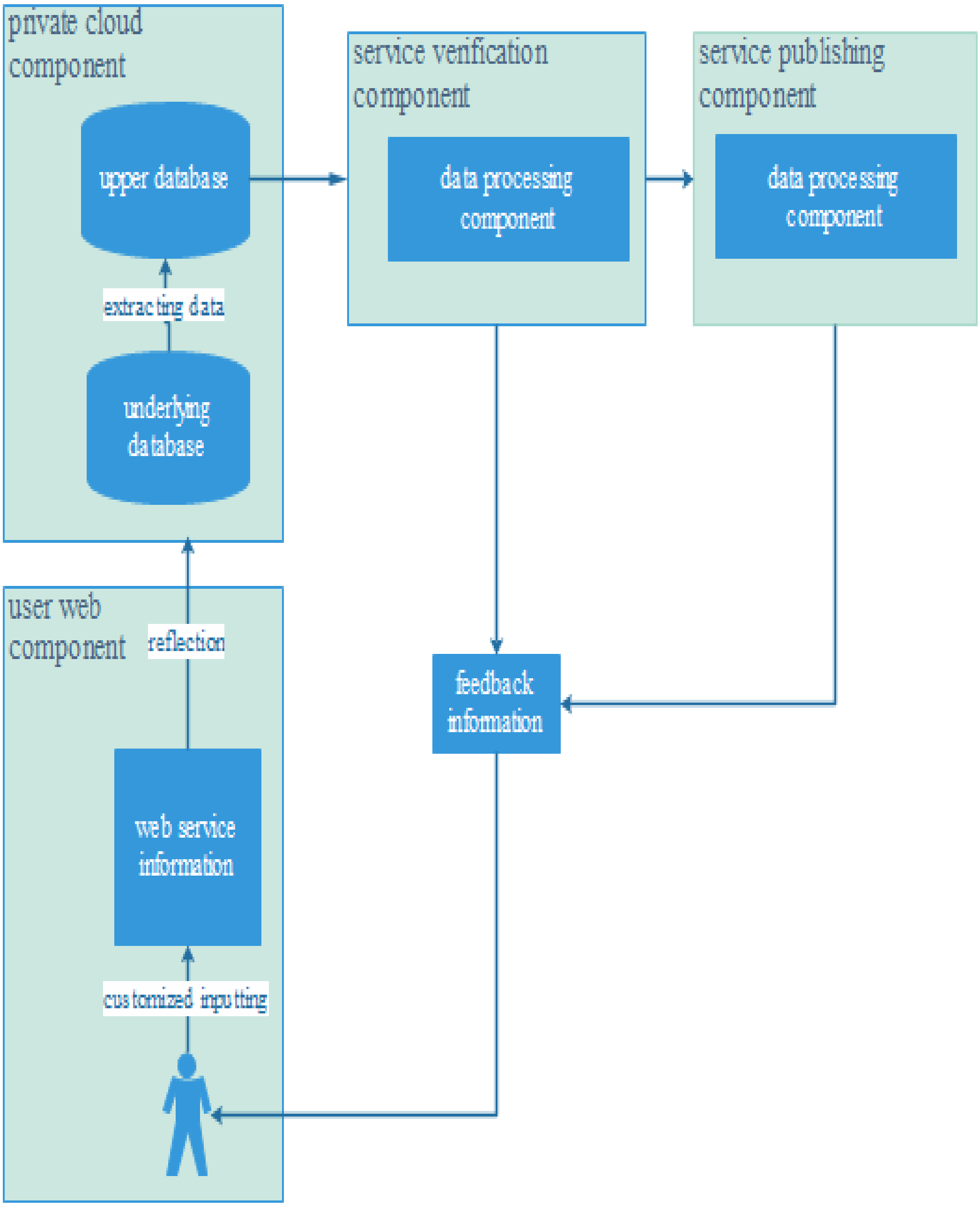


Figure 1. Components of the Model

User Web Component

The component mainly works through web pages. As one or several of its main web page appears, this component is responsible for obtaining the user's individual needs, providing user the appropriate option and input operation. In addition, this component show the feedback information to user. Also, user web components will be invoked in the course of constructing the service. The following is the performance of the various functions at different stages of the constructing process out of service.

Function 1 : Build Data Tables and Do Data Entry Operations

User describes the table names, column family names, column names *etc*. User web component generates a data table description file dynamically. The data table description file named “XML_table” which is the basis for creating a table in database. By the way, we used HBase among our private cloud. After the description of data table, a same table will be created in the database which is transparent to the user. Then the user should submit his or her own data in accordance with the structure of the table. These data will be the possible resource for constructing a web service.

Function 2: Service Information and Data Processing Type

The service information is divided into two: a part is automatically generated by obtaining user registration information, such as user name, user ID, *etc*; The other part is entered by user, such as service type, service description, key words, etc. Especially， user should make a choice for the final processing way. User can customize his or her service to be presented in chart, Web GIS or some other ways. For example, some users just want to calculate some data and save the result in a file, then the file is a possible way which could be customized. And these is stored in the file named “XML_describe”.

Function 3: Data Reflection

Select the required data. All the user’s data is stored in the private cloud, but the data used ultimately may just only part of the whole. The data used finally is determined by the user. Data reflection helps user to select the data will be used, and then finish the reflection between the selected data and the item they belonged to in the final processing. Specifically, if we choose Web GIS for the final type, the reflection is mainly between the column names and the items in the map. And this relationship of the data reflection is stored in a file named “XML_reflect” which is the basis of data extracting from the underlying database to the upper database.

Data Flow

In user web component, we get “XML_table”, “XML_describe”, “XML_reflect” through mutual cooperation with other components. Next we come to interpret the data flow through the figures.

XML_table, as the basis of creating data table, is automatically generated by the user's operation and then delivered to private cloud component. Figure 2 depicts the data flow from user web component to private cloud component.

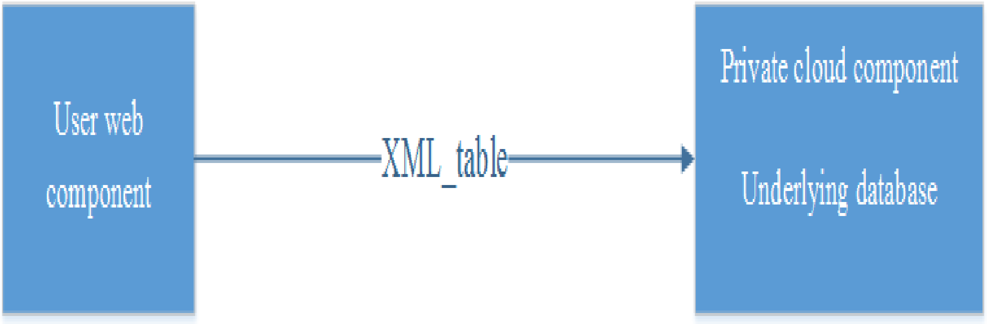


Figure 2. Data Flow from User Web Component to Underlying Database

XML_describe stores the description of the web service. Figure 3 depicts the data flow from user web component to data processing component.

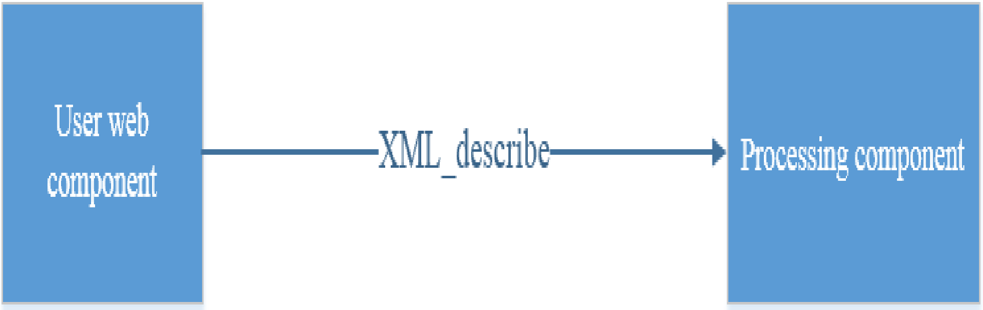


Figure 3. Data Flow from User Web Component to Processing Component

XML_reflect, as the basis of extracting data among the private cloud component. Figure 4 depicts the data flow from user web component to the private cloud component.

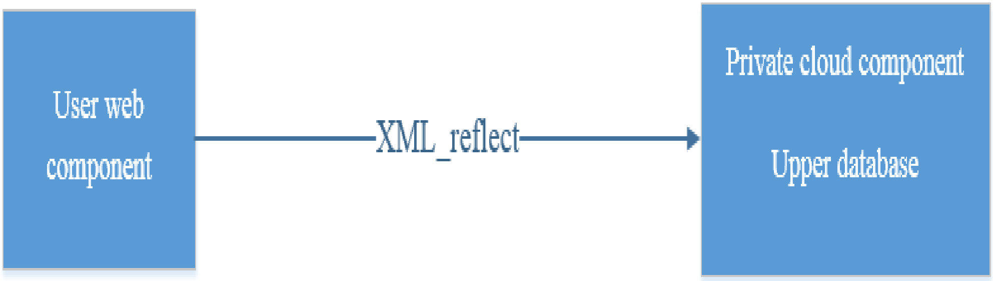


Figure 4. Data Flow from User Web Component to Upper Database

4.1. Private Cloud Component

This component is used to store user data. We provide enough private cloud space for users. When constructing a web service, the data stored in the private cloud will be the data source. But his or her own data is not the only data source. Users can invoke other available data as long as the permission to them is obtained.

This component is divided into two: the underlying database and the upper database. Through this structure, not only we improve the accuracy of the data usage but ensure the security and integrity of the data in underlying database.

Function 1

The underlying database stores the whole data from user, and it is also the data source of the upper database. XML_table here is used to create a true data table in HBase.Specific generation process will be described in detail later in the experimental section.

Function 2

The upper database stores the data required during constructing a web service. Maybe the data here is just a little part of the underlying database. The data will be extracted from the underlying database automatically according to XML_reflect, which is just as Figure 5 depicts. The data will be the source for final data processing, which is just as Figure 6 depicts.

Data Flow

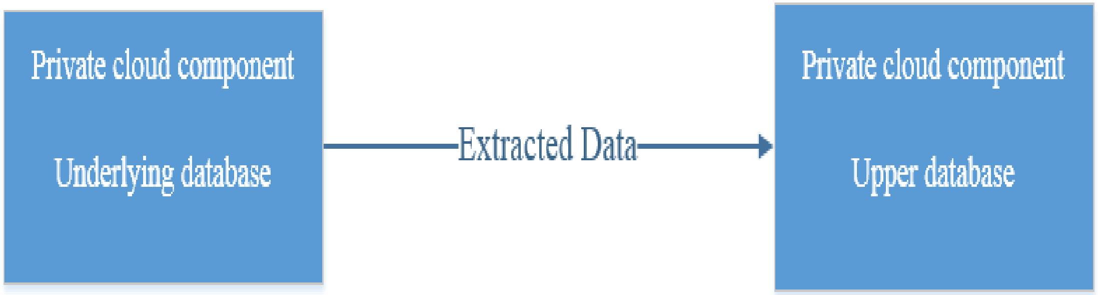


Figure 5. Data Flow from Underlying Database to Upper Database

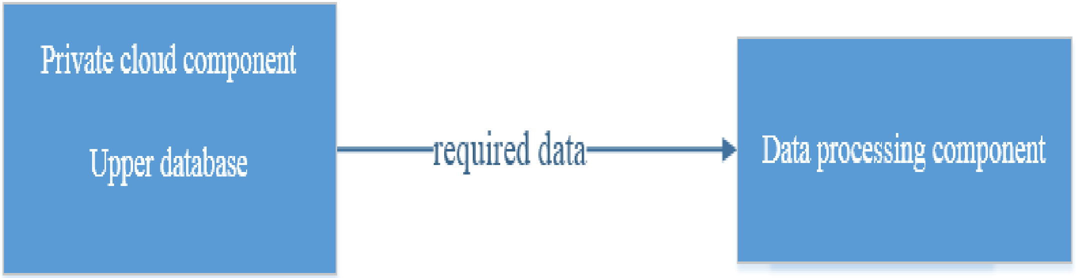


Figure 6. Data Flow from Upper Database to Data Processing Component

4.2. Web Service Verification Component Function

This component helps the background administrator to verify the legitimacy and feasibility of the web service constructed by user. The legitimacy verification will deny the construction of web service for illegal data. The feasibility verification will focus on whether the data can be processed properly. During this verification, a little part of the data will be tested.

Web service verification component invokes the interface of the data processing component to deal with the data. Then the administrator will give feedback information to user web component whatever it is successful, just as Figure 7 depicts. If there is no problem left, administrator will start web service publishing component to finish the construction, just as Figure 8 depicts.

Data Flow

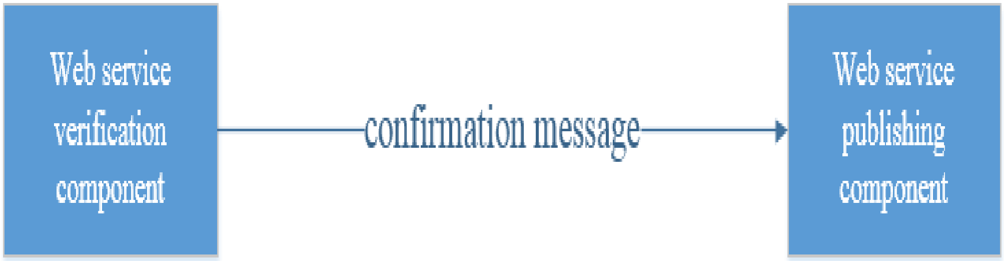


Figure 7. Data Flow from Web Service Verification Component to Web Service Publishing Component

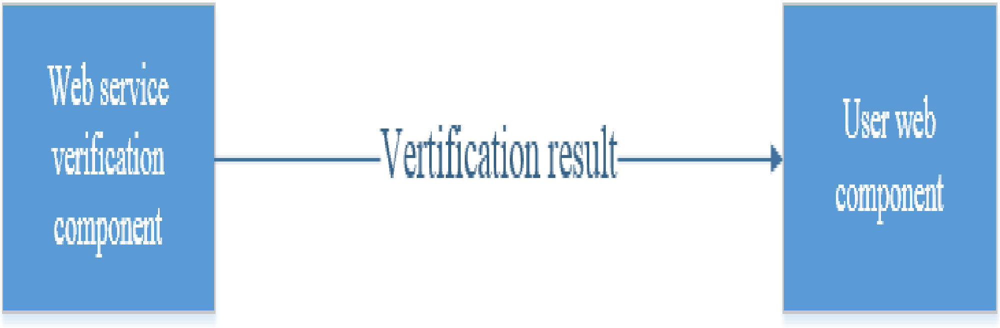


Figure 8. Data Flow from Web Service Verification Component to User Web Component

4.3. Web Service Publishing Component

Function

After receiving the positive confirmation information from web service verification component, this component will invoke the data processing component to finish the constructing. Data processing component is invoked two times if the service goes well. The difference between them is mainly about the amount of data. When invoked by the publishing component, the data processing component will deal with all the required data rather than a little part during the verification. Finally, a publishing result information will be sent to user web component which is as Figure 9 depicts.

Data Flow

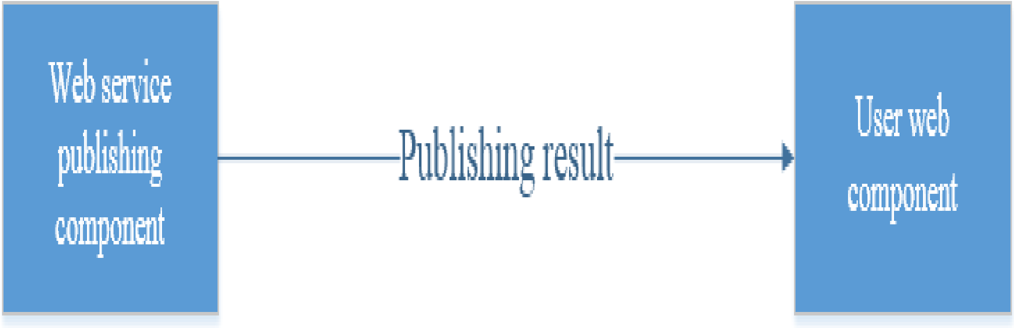


Figure 9. Data Flow from Web Service Publishing Component to User Web Component

4.4. Data Processing Component

Function

This component is the common part in both web service verification component and publishing component. The data flows are depicted as Figure 10 and Figure 11. It processes data in different ways according to the information stored in XML_describe.

Data Flow

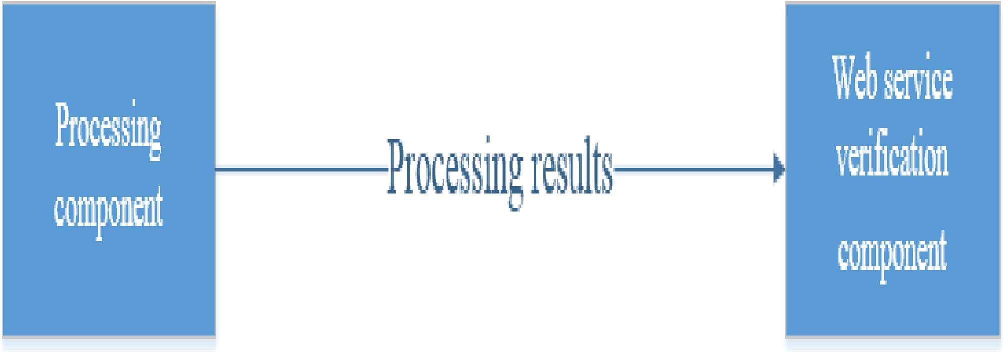


Figure 10. Data Flow from Processing Component to Web Service Verification Component

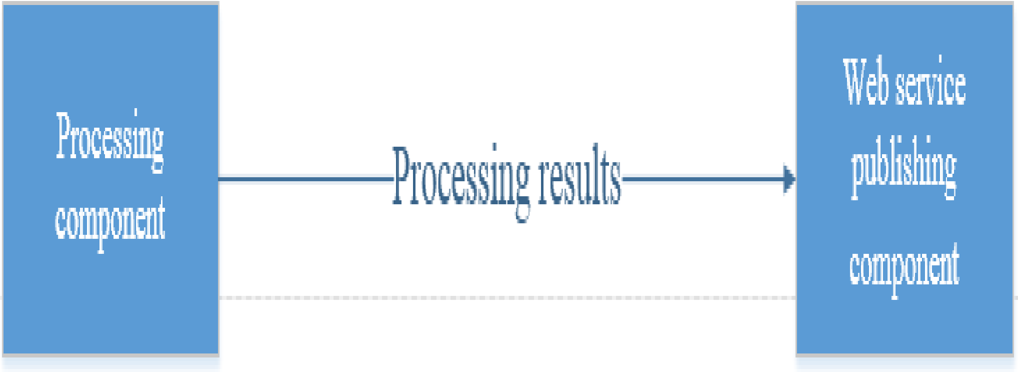


Figure 11. Data Flow from Processing Component to Web Service Publishing Component

5. Experiment

We use the existing cloud platform [15,16] which is constructed in our lab as the foundation in the experiment. And the platform is mainly about Chinese minority villages information. It can show data by latitude and longitude on a map to help with analyzing.

Our test data is associated with Chinese minority villages. In each group of villages data will include the latitude and longitude information in order to show the result in the map using Web GIS.

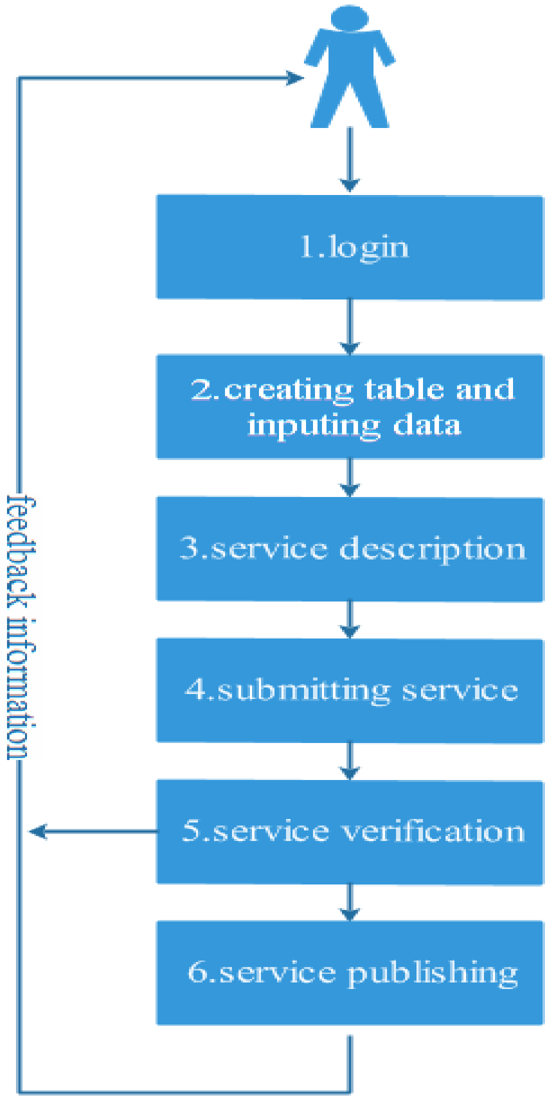
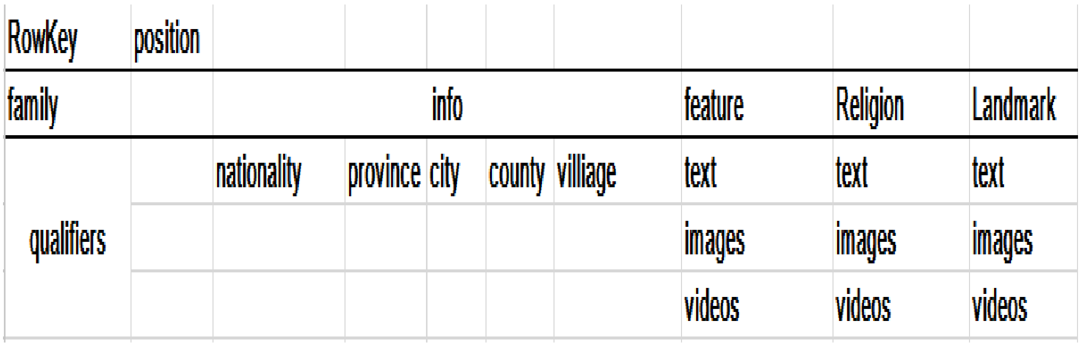


Figure 12. Main Steps of the Experiment

The main six steps are depicted as Figure 12.: after logging in the system successfully, user need to create a data table and input data. Then a service description is a must. The last thing users should do is just submission. The following works are handed to the model. Administrator will do the service verification and service publishing. What the users have to do is just waiting the feed back information.

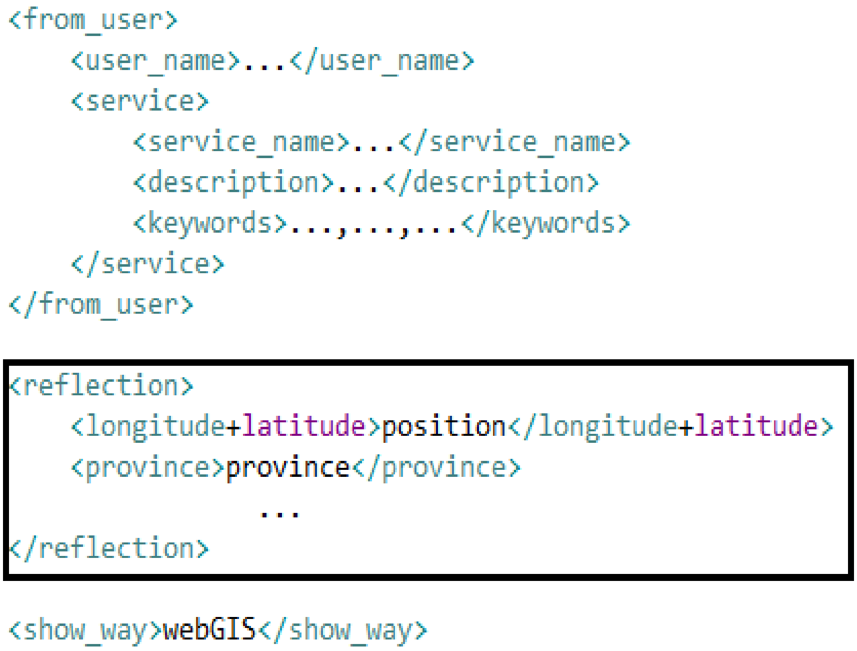
Next several steps and some specific structures will be discussed. In step 2, the model create data table in HBase [17,18,19] according to XML_table. In simple terms, the content of XML_table is all about the table structure. A specific table structure is indicated as Figure 13.



RowKey	position								
family		info					feature	Religion	Landmark
qualifiers		nationality	province	city	county	villiage	text	text	text
							images	images	images
							videos	videos	videos

Figure 13. Specific Table Structure

Then the reflection between the data source and the last map items according to user's operations and XML_table is finished in step 3 and step 4. The reflection information which are pairs of key-values is stored in the newly generated file XML_reflect. The overall structure of this file as Figure 14. For example, the key "longitude+latitude" is an item in the final map, but the value "position" is a true column name of the data table.

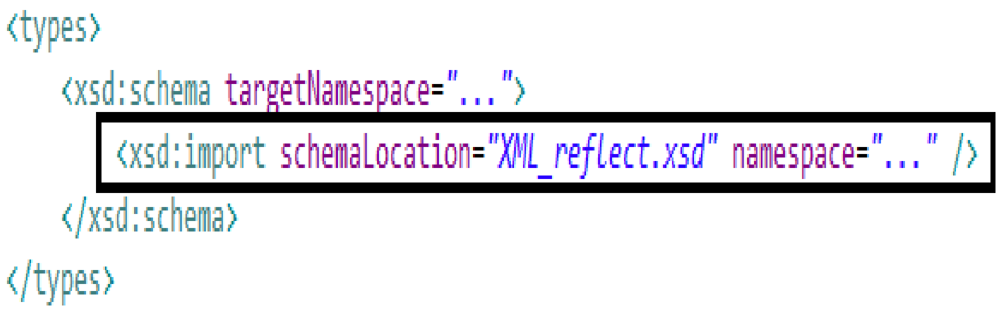


```
<from_user>
    <user_name>...</user_name>
    <service>
        <service_name>...</service_name>
        <description>...</description>
        <keywords>...,...,...</keywords>
    </service>
</from_user>

<reflection>
    <longitude+latitude>position</longitude+latitude>
    <province>province</province>
                ...
</reflection>

<show_way>webGIS</show_way>
```

Figure 14. Structure of XML_Reflect

Next the administrator will verify the web service by extracting data from the underlying database to the upper database and processing the data in a customized way in the light of these xml files in step 5. Finally, all the required data will be demonstrated in the map and feedback information will be passed to user.

After the WSDL file is generated by conventional tools and in order to improve the integrity of the WSDL document, we brought XML_reflect in by "import" statement as Figure 15 depicts.



```
<types>
    <xsd:schema targetNamespace="...">
        <xsd:import schemaLocation="XML_reflect.xsd" namespace="..." />
    </xsd:schema>
</types>
```

Figure 15. Using “import” Statement in WSDL File

6. Conclusion

In this paper, in order to allow individual users can easily construct their own web services rely on their own data source, we propose a model which just need users to do some simple configuration. The experimental results indicate: from the user's point of view, he or she can easily construct a web service based on his or her own data through simple web operations, which achieves the purpose of the model. After the construction of service, users of this platform will become a service provider who can provide services based on its own data to the other visitors. So these visitors can learn about minority villages of information from this service. Besides,we selected the test data set includes not only the text, but also a lot of videos, and the results display correctly according to the latitude and longitude through Web GIS.

7. Future Work

Currently, we only verified the feasibility of single-user service constructing model. Further more, we will achieve multi-user mode on this basis, the increasing number of users and services will help to enhance the service constructing model.

But we just discussed the process of a web service constructing. To enhance the robustness of the model, we should take service registration[20], service discovery [21,22,23], service retrieval into account. Therefore a stronger service engine will be designed in the future work.

Acknowledgments

I should like to acknowledge with deep gratitude the assistance and guidance given to me by my tutor Yu Weng. And the experiment would not have been possible without the support and assistant from all the colleagues in our lab. This paper is supported by “Construction of a comprehensive database of Development and a information service platform for Minority Areas”, “State Ethnic Affairs Commission-Leading talents supported projects”, “Ministry of Education of the People's Republic of China-Supported by Program for New Century Excellent Talents in University”.

References

[1] L. Zhou, V. Varadharajan and M. Hitchens, “Achieving Secure Role-Based Access Control on Encrypted Data in Cloud Storage”, Information Forensics and Security, IEEE Transactions, **(2013)**.

[2] A. C. Squicciarini, B. Carminati and S. Karumanchi, “Privacy aware service selection of composite web services invited paper”, Collaborative Computing: Networking, Applications and Worksharing (Collaboratecom), 9th International Conference, **(2013)**.

[3] G. Suciu, E. G. Ularu and R. Craciunescu, “Public versus private cloud adoption — A case study based on open source cloud platforms”, Telecommunications Forum (TELFOR), 20th , **(2012)**.

[4] S. Le, D. Hai and J. Ashraf, "Survey of Service Description Languages and Their Issues in Cloud Computing", Semantics, Knowledge and Grids (SKG), Eighth International Conference, **(2012)**.

[5] N. Parimala and A. Saini, Web service with criteria: Extending WSDL, **(2011)**.

[6] M. Crasso, J. M. Rodriguez, A. Zunino and M. Campo, Revising WSDL Documents: Why and How, **(2010)**.

[7] C. Mateos, M. Crasso, A. Zunino and J. L. O. Coscia, Revising WSDL Documents: Why and How, Part 2, **(2013)**.

[8] T. G. Banage S. Kumara, I. Paik and W. H. Chen, "Web-Service Clustering with a Hybrid of Ontology Learning and Information-Retrieval-Based Term Similarity", Journal of Convergence Information Technology(JCIT), **(2013)**.

[9] T. Fortis and V. Negru, "Service Brokering in Cloud Governance", Munteanu, V.I., Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 14th International Symposium, **(2012)**.

[10] D. A. D'Mello, S. Salian, N. S. Sahana, C. Rodrigues and C. D'Souza, "A Service Broker Based Architecture for Effective Discovery and Automatic Publishing of Internet Web Services", 2nd IEEE International Conference on Parallel, Distributed and Grid Computing, **(2012)**.

[11] G. S. Gill, A. Wadhwa and A. Jatain, "Cloud Computing: A New Age of Computing", Advanced Computing & Communication Technologies (ACCT), Fourth International Conference, **(2014)**.

[12] D. Kallergis, C. Papacharalampou, K. Chimos, T. Chavakis and C. Douligeris, "A Hybrid Cloud Computing Approach for Managing Spatial Data: A Case Study for Water Resources in Greece", Computing for Geospatial Research and Application (COM.Geo), Fifth International Conference, **(2014)**.

[13] E. B. Chieng, "Performance and security issue on open source private cloud", Electrical Engineering Congress (iEECON), **(2014)**.

[14] Z. C. Feng, D. H. Fang and Q. Q. Ye, "Design and Implementation of Hybrid Cloud Computing Architecture Based on Cloud Bus", .Mobile Ad-hoc and Sensor Networks (MSN), IEEE Ninth International Conference, **(2014)**.

[15] K. Surya, M. Nivedithaa, S. Uma and C. Valliyammai, "Security issues and challenges in cloud", Green Computing, Communication and Conservation of Energy (ICGCE), International Conference, **(2013)**.

[16] C. Bin, X. Feng, Y. Feng and Z. W. Huan, "Research and application of migrating legacy systems to the private cloud platform with cloudstack", Automation and Logistics (ICAL), IEEE International Conference, **(2012)**.

[17] D. Han and E. Stroulia, "A three-dimensional data model in HBase for large time-series dataset analysis", Maintenance and Evolution of Service-Oriented and Cloud-Based Systems (MESOCA), IEEE 6th International, **(2012)**.

[18] D. H. Jie, J. Y. Hui, C. Y. Dong and Y. Tan, "Distributed storage of network measurement data on HBase", Cloud Computing and Intelligent Systems (CCIS), IEEE 2nd International Conference, **(2012)**.

[19] H. Ochiai, H. Ikegami, Y. Teranishi and H. Esaki, "Facility Information Management on HBase: Large-Scale Storage for Time-Series Data", Computer Software and Applications Conference Workshops (COMPSACW), IEEE 38th International, **(2014)**.

[20] S. Silcher, J. Konigsberger, P. Reimann and B. Mitschang, "Cooperative service registries for the service-based Product Lifecycle Management architecture", Computer Supported Cooperative Work in Design (CSCWD), IEEE 17th International Conference, **(2013)**.

[21] K. Abrougui, A. Boukerche, R. W. Pazzi and M. Almulla, "A Scalable Bandwidth-Efficient Hybrid Adaptive Service Discovery Protocol for Vehicular Networks with Infrastructure Support", Mobile Computing, IEEE Transactions, **(2014)**.

[22] C. Manike, P. P. S. Naik and K. D. Prasad, "An efficient framework for high-quality web service discovery", Software Engineering (CONSEG), CSI Sixth International Conference, **(2012)**.

[23] P. Harshavardhanan, J. Akilandeswari and R. Sarathkumar, "Dynamic Web Services discovery and selection using QoS-Broker architecture", Computer Communication and Informatics (ICCCI), International Conference, **(2012)**.

Authors

Weng Yu, he is an associate professor of College of Information Engineering, Minzu University of China. His current research interest are distribute computing and service computing.

Peng Guo, he is a full-time graduate student in College of Information Engineering, Minzu University of China and he majors in computer Science technology.