Designing Formulas for Creating a Healthcare Cloud Service Based on a Formula Calculation Platform Service

Xing Chen
Department of Information & Computer Science
Kanagawa Institute of Technology
Atsugi-shi, Kanagawa, Japan
chen@ic.kanagawa-it.ac.jp

Keiichi Shiohara
Department of Information & Computer Science
Kanagawa Institute of Technology
Atsugi-shi, Kanagawa, Japan
s1385006@cce.kanagawa-it.ac.jp

Hikaru Tazumi
Department of Information & Computer Science
Kanagawa Institute of Technology
Atsugi-shi, Kanagawa, Japan
s1021101@cce.kanagawa-it.ac.jp

***Abstract*— Cloud services provided for healthcare are easy to use and updated quickly with new knowledge therefore they are widely utilized in nowadays. A research issue here we need to solve is to provide a no-programming method for developing cloud services. The research issue comes from the background that the specialists in the area of healthcare are not also the specialist in the area of information technology. In this paper, we propose a method for creating a healthcare cloud service based on a cloud service that we developed referred to as the Formula Calculation Platform Service. The important feature of the method is that service builders only need to consider about the functions of their services. Based on the proposed method, we created a cloud service for weight control to evaluate the effectiveness of the proposed method. The effectiveness of the proposed method is confirmed by demonstration experiments.**

Keywords-cloud service; programming; platform; healthcare; weight control.

I. Introduction

Healthcare services, such as businesses-oriented health management system [1][2] and general health support services for individuals and families [3], etc., are highlighted gradually. A wide variety of services for healthcare are developed, which are utilized regardless of gender and age, etc. In order to let the services to be widely utilized, it is important to update the service constantly based on the new knowledge. When users utilize a service for healthcare at first time, they do not know whether the service suits them or not. Therefore, it is a common case that they first try it to see if it suits them or not. Depending on the needs of the users, it is important to update services continuously or to add new functions to prevent users boring the services and help spread the services to be widely utilized.

We focus our research on methods of building cloud services. We have developed a cloud service platform, referred to as a Formula Calculation Platform Service (FOCAPLAS) [4]. By using this service, without programming, cloud services can be implemented based on mathematic formulas uploaded by service providers. In the same way, using this service, cloud services can also be updated when mathematic formulas are modified.

In this paper, we represent formulas for implementing a weight control cloud service as an example showing the efficiency of FOCAPLAS. Based on the proposed formulas, a weight control cloud service is created without program coding. We also performed demonstration experiments for the created cloud service. Our experimental results made it clear that data of the service users are correctly stored, formulas, which are uploaded by service providers, are executed correctly. Therefore, we confirmed the validity of our proposed method.

The rest of the paper is structured as follows. Section II presents related researches in this research area. Section III presents the outline of FOCAPLAS. Section IV explains the basic idea of constructing the cloud service for the healthcare. After this, Section V presents an implantation example for developing a weight control cloud service. Our experimental results are also presented in this section. Technical details about communication protocols are presented in Section VI. Finally, Section VII concludes this paper.

II. Related Researches

The purpose of healthcare services is to provide healthcare related comments, or to provide healthcare related computing results based on the data of the users. From technical point of view, healthcare services can be considered as a kind of knowledge processing system.

In the research field of Information and Communication Technology (ICT), knowledge processing is an important research field. Various methods related to the knowledge processing have been proposed [5]. In particular, as the developing of ICT, a large amount of knowledge information data is distributed in cloud, research studies on knowledge information retrieval [6], and knowledge information coordination [7] are performed. In addition, the research studies are also performed on providing knowledge services [8], and knowledge sharing and recommendations [9] based on the stored data of knowledge information in servers.

As not only knowledge contents, but the calculation results are also required to be provided, services are developed, which provide the calculation results based on knowledge [10][11].

In the case that automatic calculation based on the knowledge is provided as a service, it is required to code a program and build a server to run the program. Traditionally, in order to build the server, high-ICT knowledge and high-quality computers are required. Currently, as pre-built server platforms are provided as cloud services [12][13], the hurdle for building a server has been lowered significantly. Furthermore, Integrated Development Environment (IDE) platforms are also provided as Platform as a Service (PaaS). These systems are designed for system developers focusing on the core applications.

One example of this kind of platform is the system called Zoho Creator [14] providing an IDE for creating custom Online Database Applications. Another example is the system called COSCA [15] that provides a component-based PaaS system. A famous example is Google App Engine. App Engine is designed for developers developing applications using the program language Java, Python or Go and to run the applications on Google's infrastructure [13].

We choose the platform FOCAPLAS. This platform is different from those introduced above because it is designed for developing applications by spreadsheet formulas. By using this platform, cloud services for end users can be developed by service developers who have only spreadsheet knowledge. A platform similar to FOCAPLAS is called XCute [16]. Different from FOCAPLAS, XCute only provides Microsoft Excel ® based Web service.

In this paper, we focus on how to develop a cloud service based on FOCAPLAS, which requires mechanisms of user identification and database management. The weight control cloud service is a good developing target because all those mechanisms are required. Furthermore, it is also required to have the ability of receiving formulas posted by experts who create the formulas based on their knowledge, and receiving weight data posted by end users who use the service and feedback analysis results back to the end users based on the calculation results of the formulas.

III. The Formula Calculation Platform

The Formula Calculation Platform Service (FOCAPLAS) is a cloud service building platform that we developed. By using this service, a service provider can build a cloud service by posting formula contents like posting document contents to a blog server. If the providers want to update their services, what is required is to post new formula contents.

We are building several cloud services and testing the services for validating the effectiveness of the platform. In our testing experiments, we built a computational service, a user authentication service, a social networking service and a Web Application Program Interface (API) service. Compared with the other services similar to ours, the biggest feature of our service is that it provides a mechanism by which cloud services are built based on formulas posted by providers.

Our platform is used in the way as follows. A service provider uploads formulas to the platform server. The uploaded formulas are parsed on the server and executable program codes are generated automatically. When a user of the service connects to the server, a data submission form is sent to the user from the server side, or an application-software connected to the server is called running by the server. Data submitted by users will be calculated based on the provider-uploading formulas. Calculating results will be sent back to the users directly or indirectly. Here, indirectly sending the calculating results means that they are used as a part of query to search related data from database. In the services like the weight control service, recommendation services, etc., the retrieval results, for example, comments or recommendations will be sent back to users.

In Fig. 1, the structure of the platform is illustrated. The platform is composed of a formula parser, a formula calculator, a database management system, data storages, an input/output device and interface specification interpretation equipment.

The formula calculator receives the data sent by the service-users from the input/output device, computes received data and returns the calculation results back to the users. The formula parser parses formulas sent from service providers via the input/output device, generates executable codes and stores them into the database system.

The database management system is designed to provide functions of reading and storing six different kinds of data: (1) executable codes generated based on uploaded formulas, (2) user data, (3) user registration information, (4) response data, (5) user interface information and (6) Web API information. The interface device is designed for generating input and output forms or Web APIs defined by the serviced providers in the format of html, xml, etc. and sending them to the input/output device.

A. The Formula Parser

The formula parser parses formulas posted by service providers and generates the following results: service Uniform Resource Locator (URL), input/output forms, application interfaces, and executable codes.

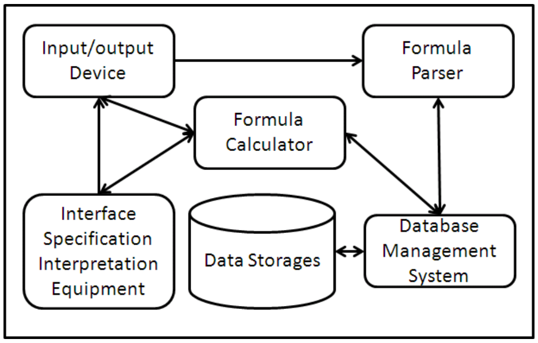

Input/output Device
Formula Parser
Formula Calculator
Interface Specification Interpretation Equipment
Data Storages
Database Management System

Figure 1. The structure of the formula Calculation Platform

Provider-posting formulas are opened to publics as cloud services. The formula parser can parse the formulas uploaded in spreadsheet, csv and text formats. Provider-posting formulas are parsed by the formula parser and converted into program codes that can be executed by the formula calculator.

B. *The Formula Calculation Service*

There are two types of formula calculation services. One is the kind of services that do not require user identification, such as “Converting Japanese calendar to the A.D. calendar”, “Loan calculation”, etc. The process of the service is performed in the way that user’s submitted data are calculated and calculation results are feedback to the users. During the calculation process, the user identification is not required.

Another one is the kind of services, which require user identification, such as “Cumulative intake of calories”, “Weight changes”. The processes of this kind of services rely on historical data submitted by users. During the service processing, when new user-submitting data is received, user’s previous submitted data is searched from the database system, and the current data is calculated together with the previous data.

In this paper, we discuss the latter case, in which user identification is required during the process of services. In our system, user IDs and user IDs associated data can be obtained from the database. However, we still need to develop a mechanism that connects formulas and user identification. In order to provide the healthcare service, we represent formulas for writing user-submitting data into the database and reading them back from the database based on user IDs.

C. *The Database Management System*

The database manage system manages a public database and user personal databases. The public database is accessed without the requirement of user identification. The user identification ID is required when a private database is accessed.

The database management system provides two kinds of management methods managing the private databases. The first kind of the method is to manage the database in which user IDs are used as the keywords during the database accessing. Another method is to manage individual user’s databases. In this method, user’s personal data are separately stored into different databases. When a user’s personal database is accessed, an identification key is required.

We select the latter method to provide the healthcare service. A user’s personal database is created when the user ID is created. If a user’s personal database is not accessed during a period of times, it will be deleted. We will represent the formulas for personal database creation and deletion in section V.

IV. Creating a Weight Control Cloud Service Based on the Formula Calculation Platform

In this section, we first describe the issue that have to be solved for creating the weight control cloud service. Then, we will represent our proposal to solve the issue after describing the related techniques.

A. *The Issue for Creating the Weight Control Cloud Service*

The previous user’s submitted data are important when the formulas are executed during the process for providing the weight control service. However, the mechanism is required to connect user IDs, correlated to the submitted data, with the formulas uploaded from service providers. With this mechanism, the user’s data can be identified under user IDs. In this way, we can find whether they are submitted by multiple users in a same day or submitted by a same user in different days. In addition, as user IDs are not filled in formulas, this mechanism helps to search user’s submitted data connected to the user IDs from the database system.

B. *Obtaining the User Identification Information*

The data submitted to the platform server contains user identification information. Although user identification information is not contained in the data submitted to the server at the first time, it can be generated in the server side and saved in the user side. Thereby, the user identification information can be sent together with the data submitted from the next times. The user identification information can be stored in the user side in directly or indirect ways. Another method is to let a user fill a form with identifying information, such as user’s login name and password. When the user identification information is used to generate a user ID, user-submitting data can be read and stored in the database under user IDs, which is used as the query keys. That is, user’s previously submitted data can be obtained based on the generated user IDs.

C. *Database Functions of the Platform*

Different from the formula calculator, the database management system has the ability to read and write user IDs and user data, from and to the data storages. That means user data can be processed individually for each user. Utilizing this ability, it is possible to perform calculation with the previously submitted data, which can be obtained through the database management system, by sending them to the formula calculator. In the platform, formulas posted by the service provider are executed as different threads. That means user data will not be mixed during the formula execution processes.

D. *Our Proposed Method*

We define the keyword that appears in the database as “*k”* and its related database operation function as “*Sel*”.

$$d_k \underset{Key=k}{=} Sel(Data) \tag{1}$$

In formula (1), the keyword is defined as “*k*”, the field of the keyword is defined as “*Key*” and the data field is defined as “*Data*”. Data stored in the database is defined as “d_k” and the operation function for reading data from the database is defined as “*Sel*”.

An inverse function of "*Sel*", *Sel^{-1}* is defined for storing user data into the database.

$$Data \underset{Key=k}{=} Sel^{-1}(d_k) \tag{2}$$

During the processing of formulas (1) and (2), "*k*" is used as the keyword for reading and writing data, from and into the database. Formula (1) is used to read previously submitted data, "d_k", from the database and formula (2) is used to store new submitted data to the database.

We also defined a formula to separate user identification information "*i*" and user data "*D*" from the user-submitting data "D_i".

$$\langle D, i\rangle = Sep(D_i) \tag{3}$$

In formula (3), the data submitted by user "*i*" is defined as "D_i". During the processing of formula (3), user identification ID "*i*" and user data "*D*" are separated from "D_i".

As user identification information "*i*" can be used as the keyword for reading/writing data from/to database, it is possible to get the previous submitted data associated with the user ID. That is, data "*D*" can be read from or written to the database under user ID. Therefore, service providers only need to fill the data "*D*" in their formulas. User management is not required to be considered by the service providers.

We utilize the mechanisms of the formula calculator and the database system of the Formula Calculation Platform as the developing platform. We created the Weight Control Service to verify advantages of the platform. Based on our experiments, it is clear that cloud service with user identification can be developed without program coding. Developing time for the service is greatly reduced.

In order to protect user privacy, we selected the anonymous authentication. In our method, when a user connects our service at the first time, an anonymous user ID is created in the server side and sent to the user. The anonymous user ID is opened for mobile terminal application development. The system is also designed to accept manual inputted user ID. We also designed a mechanism to process data omissions based on the user's previously submitted data. In the service, graphs of the weight changes are generated and sent to users as it is known that showing the graphs of weight changes is helpful for weight control [17].

V. Service Creation and Experiments

A. Formulas for Creating the Service

We performed demonstration experiments to confirm the feasibility of the proposed method. We prepared two types of interfaces, a Web API interface for mobile devices and a HTML interface for general-purpose Web browsers. In order to generate the anonymous user ID, we create a vector, **V**, with 26 English uppercase characters.

$$\mathbf{V} = (A, B, \cdots, Z) \tag{4}$$

Each element of the vector is defined by the formula *V(i)*. In the formula, "*i*" is an integer in the range from 1 to 26. For example, *V(1)* indicates the letter "A" and *V(26)* indicates the letter "Z". An anonymous user ID is defined by a vector with "*n*" elements. Each element is a randomly selected English uppercase letter. We use the following formula to generate anonymous user ID, where rand(1…26) is a random number generator generating a number from 1 to 26.

$$ID = \sum_1^n v(rand(1\cdots 26)) \tag{5}$$

We set "*n*" to 6 in our service. User's daily measured weight data are stored in the database system. In many cases, it is impossible letting a user input weight data every without omission. Various cases can be considered which will result to data omissions. For example, during business trips, it may be impossible to submit weight data. Or a user just does not want to submit the data because of overtime working. Considering these cases, data omissions are irregular. We use Newton polynomial [18] to interpolate the omitted weight data as it has a characteristic that the interpolated data can be reused for another interpolation. For the omitted dates, we use the linear interpolation [18] to interpolating the omitted dates with 1-day interval. Fig. 2 shows an interpolation result. In the figure, weight data are submitted on July 1, 2 and 7. The omitted dates on July 4, 5 and 6 are interpolated.

As shown in Fig. 2, in order to reduce the amount of computation on interpolation calculation, the service is designed to store data up to 7 days.

If the user identification ID, which is separated from the user-submitting data, is not stored in the database, it will be stored into the database as a new ID. We define the user identification field as *ID*, and the user ID stored in the database as ID_d. We use the following formula to determine whether the user ID is stored in the database or not. If it is not stored in the database, ID_d is defined as $ID_d = 0$.

$$ID_d \underset{Key=ID_k}{=} Sel(ID) \tag{6}$$

If the user identification ID, ID_k is not stored in the database, that is, $ID_d = 0$, we use the following formula to store it into the database.

$$ID \underset{Key=ID_k \cap ID_d=0}{=} Sel^{-1}(ID_k) \tag{7}$$

In the formula, "∩" is defined as the logical product.

We use formula (8) to store user's weight data into the database, where W_k is the user's weight data and *W* is the field of weight data.

$$W \underset{Key=k}{=} sel^{-1}(W_k) \tag{8}$$

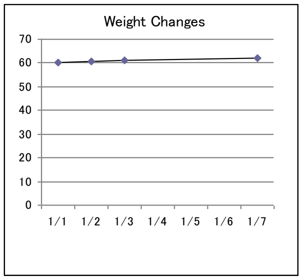


Figure 2. Interpolation result for 3 days

Formula (9) is used to read the weight data from the database.

$$W_k \underset{Key=k}{=} Sel(W) \tag{9}$$

B. Formulas for the Database Operations

Formulas for database creation and deletion are supported by the Formula Calculation Platform. In order to create a database, a field name vector, **N**, and a field character vector, **C**, are required. The database creation formula is defined as "*Cre*".

$$Cre(\mathbf{N}, \mathbf{C}, name) \tag{10}$$

In the formula, "*name*" is the name of the created database. When the vector **N** or **C** is an empty vector, the database "*name*" will not be created.

The user personal database is crated in the case that the user identification ID, ID_k is not stored in the ID database (ID_d=0). As we define all the user personal databases having the same fields, we define two common vectors, $\mathbf{N}_{cm}$, $\mathbf{C}_{cm}$, for the database creation. After the user identification ID, ID_k, is created, the user personal database will be created based on the following formulas, (11), (12) and (13).

$$\mathbf{N} \underset{ID_d=0}{=} \mathbf{N}_{cm} \tag{11}$$

$$\mathbf{C} \underset{Id_d=0}{=} \mathbf{C}_{cm} \tag{12}$$

$$Cre(\mathbf{N}, \mathbf{C}, ID_k) \tag{13}$$

Based on the formulas (11) and (12), for a user ID that is already generated and stored in the ID database, the vectors **N** and **C** are the empty vectors. Therefore, new user database will not be created.

The formula for database deletion is defined as "*Del*". A personal databases will be deleted which are not accessed more than 7 days. In the "*Del*" formula, formula (14), a database name, *name*, is required.

$$Del(name) \tag{14}$$

In the formula (14), if *name* is an empty string, the function will not be executed.

We use the following formula to control the delete operation. In the formula, the user ID log is used. When an ID is not active more 7 days, the ID will be sent to the string variable, *dn*. When the no-empty string variable is sent to the formula *Del*, the database, which name is stored in the variable *dn*, will be deleted.

$$dn \underset{ID_k|\text{unactive log}=7}{=} ID_k \tag{15}$$

The personal database is deleted by using the following formula.

$$Del(dn) \tag{16}$$

C. Checking the Operation of the Created Cloud Service

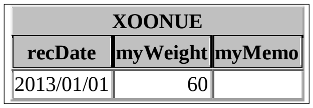
The operations of reading and writing user identification ID, and user's submitted weight data are check. The execution of the uploaded formulas for weight control is also checked.

In our experiments, we first checked whether a new ID key is created for the beginning user or not, and whether the new ID key is stored in the database or not. After that, we confirmed whether the user-submitting data is stored under the ID key into the database or not. The operation check result is shown in Fig. 3. We confirmed that a new ID key is created for the beginning user during the data submitting. For the subsequent data submitting, the user identification information is separated from the submitted data and stored into the database.

We also confirmed that new submitted weight data are correctly stored into the database under the already existed ID key. Fig. 4 is the result of the operation check.

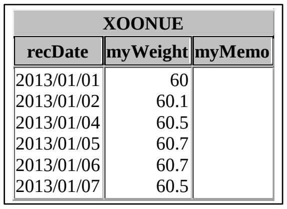
We confirmed interpolation for the data omissions. Fig. 5 is the result of the operation check. As shown in Fig. 5, there is a data omission on January 3. The interpolation works correctly as shown in the figure. We also check the operation of the interpolation for the continuous data omissions. Fig. 6 is the result of the operation check. There are data omissions on January 3, 4 and 5. As shown in the figure, the interpolation works correctly.

We verify the operation of the service used by mobile terminals. Fig. 7 is the result of the operation check. The operation check was carried out by using Android mobile cell phones.



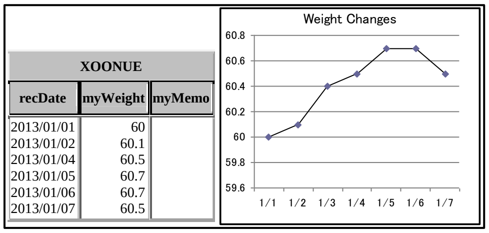
XOONUE		
recDate	**myWeight**	**myMemo**
2013/01/01	60	

Figure 3. The operation check result on new ID key generation



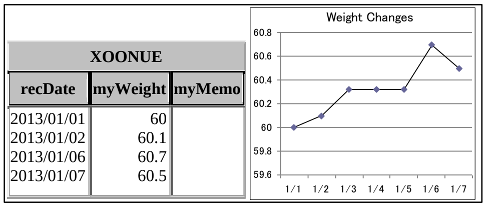
XOONUE		
recDate	**myWeight**	**myMemo**
2013/01/01	60	
2013/01/02	60.1	
2013/01/04	60.5	
2013/01/05	60.7	
2013/01/06	60.7	
2013/01/07	60.5	

Figure 4. New weight data are stored under the ID key



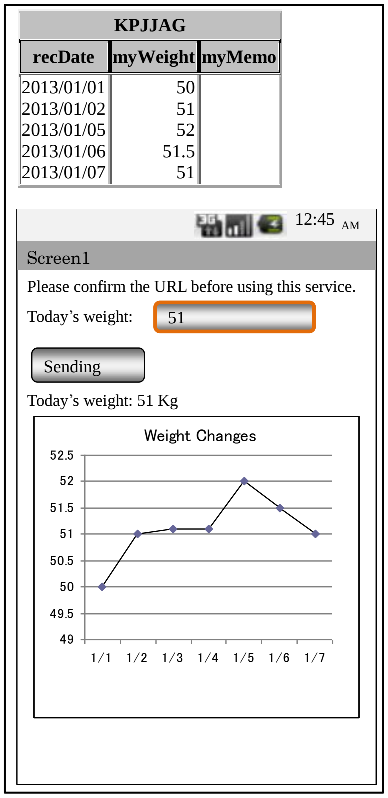
XOONUE		
recDate	myWeight	myMemo
2013/01/01	60	
2013/01/02	60.1	
2013/01/04	60.5	
2013/01/05	60.7	
2013/01/06	60.7	
2013/01/07	60.5	

Figure 5. Interpolation for the one day omission



XOONUE		
recDate	myWeight	myMemo
2013/01/01	60	
2013/01/02	60.1	
2013/01/06	60.7	
2013/01/07	60.5	

Figure 6. Interpolation for the continuous data omission



KPJJAG		
recDate	myWeight	myMemo
2013/01/01	50	
2013/01/02	51	
2013/01/05	52	
2013/01/06	51.5	
2013/01/07	51	

Please confirm the URL before using this service.

Today's weight: 51

Today's weight: 51 Kg

Figure 7. The operation check on an Android mobile cell phone

VI. TECHNICAL DETAILS ABOUT THE COMMUNICATION PROTOCOLS

In this section, technical details about the communication protocols are presented. In the Formula Calculation Platform (FOCAPLAS), themes are provided to the cloud service providers. Two different types of themes are provided. The first one is designed for Web browsers. The second one is designed for Web API application developers. Cloud service providers select the themes that are suitable to their services. Based on the selected themes, service providers design and upload their formulas. By using the selected themes, cloud services are performed in the following steps:

(1) Users' service requests are represented as arguments of program functions, which are generated based on the providers' functions.

(2) The program functions are called and executed according to the service requests.

(3) Return values of the program functions are formatted based on the design of the selected theme and sent back to the cloud service users.

In the FOCAPLAS, we use the Hypertext Transfer Protocol (HTTP) as the data communication protocol. We have designed three kinds of themes: (1) using GET method only, (2) using both GET and POST methods and (3) using GET, POST and PUT methods. The themes including JavaScript codes are also designed and created.

In the FOCAPLAS, an assignment operator is designed and created. Request messages are assigned to the arguments of the program functions by the assignment operator. Response messages are generated based on the return values of the program functions in the format defined in the theme, which is selected by the service provider.

To send back response messages back to the Web API applications, character strings are enclosed by tags, which indicate start and end positions of a string. According to these tags, a character string, which will be displayed in a label, can be extracted from the return message. Furthermore, an image, which will be displayed in a canvas, can also be extracted from the return message by using the tags.

The tags are user definable. We defined three kinds of tags for the weight control cloud service. The first kind of the tags is used for the string display, which is used to extract the character string that will be displayed in a label. This kind of the tags is referred to as "tags for label."

The second kind of tags is used to extract the graph URL from the return message, which shows the daily changes of weights. This kind of tags is referred to as "tags for information resource."

The third kind of tags is used to extract user identification strings from the return message, which is used for sending user ID's associated data to the server. This kind of tags is referred to as "tags for ID."

The tags used in the weight control service, with which the operation check result is shown in Fig.7, are summarized as follows:

(1) Tags for label:
Start tag: LabelStart
End tag: LabelEnd
Example:
LabelStart
Today's weight: 51Kg
LabelEnd

(2) Tags for information resource:
Start tag: ImageStart
End tag: ImageEnd
Example:
ImageStart
http://weitght.control.jp/AIEGOP.gif
ImageEnd

(3) Tags for ID:
Start tag: CodeStart
End tag: CodeEnd
Example:
CodeStart
KPJJAG
CodeEnd

VII. Conclusion and Future Work

In this paper, we represented a method for creating cloud services based on our developed Formula Calculation Platform Service. In the method, we proposed formulas for separating user identification information, and formulas for reading and writing user-submitting data under the user identification information. We created a weight control cloud service to evaluate the efficiency of the proposed method. Our evaluation experiments were performed on three subjects.

The first subject is to evaluate the execution of the uploaded formulas. In our weight control service, we proposed individual personal-based formulas. The cooperative operation of the proposed formulas and the database manager system is a key point to confirm the effectiveness of the proposed method. We confirmed that the proposed formulas are executed correctly based on our experimental results.

The second subject is to confirm the operation of the user identification ID. We confirmed that new keys are created to the users who first used the service. We also confirmed that user identification IDs are correctly separated from the user-submitting data into the user identification ID and the user weight data. The uploaded formulas correctly worked with the server network management system. Based on the experimental results, our proposed formulas are correctly executed.

The third subject is to test the uploaded formulas working as a cloud service. It is extremely important that these formulas work together as a system providing the required cloud service. We confirmed that the weight control cloud service that we created is running correctly.

Based on the experimental results, we confirmed the effectiveness of our proposed method. That is, we have reached our goal of developing a cloud service for end users using formulas only. Our method is different from the other methods [13][15][19] because our method is suitable for the cloud service builders who have less ICT techniques. We demonstrated that cloud services that need the mechanisms of user identification and database management can also be developed by using formulas presented in this paper. By using FOCAPLAS, the overhead of developing cloud applications is greatly reduced and the hurdle for developing cloud applications is also lowered significantly.

As our future work, we will present more applications with more technical details. As a quickly developing technology, many new method and techniques are presented in the cloud computing research area. We are taking a great interest in the relative research works, especially in the area of PaaS and mobile applications working together with cloud computing.

References

[1] NEC Corporation, “Health management cloud service”, http://www.nec.co.jp/solution/cloud/service/saas_common/healthcare.html, (in Jaspanese) (accessed 2013-02-12).

[2] NTT Resonant Inc., “A cloud-based healthcare services: ‘goo body log’ ”, http://pr.goo.ne.jp/detail/1670/, (accessed 2013-02-12).

[3] Ryobi Systems, “Health Medical Recorder (General health managing and supporting system)”, http://rcs.ryobi.co.jp/solution/by-company/rs-health-business/item/40-kenko-karte.html, (in Japanese) (accessed 2013-02-12).

[4] Chen Lab., “FOCAPLAS”, http://www.chen.ic.kanagawa-it.ac.jp/focaplas/index.html, (in Japanese), (accessed 2013-05-19).

[5] Maryam Alavi and Dorothy E. Leidner, “Review: Knowledge management and knowledge management systems: Conceptual foundations and research issues”, MIS Quarterly, Vol. 25, No. 1, Mar., 2001, pp. 107-136.

[6] Yasushi Kiyoki, Takashi Kitagawa, and Kayoko Kurata, “An Adaptive Learning Mechanism for Semantic Associative Search in Databases and Knowledge Bases”, Information Modelling and Knowledge Bases, VIII, IOSW Press, May., 1997, pp. 345-360.

[7] Takafumi Nakanishi, Koji Zettsu, Yutaka Kidawara, and Yasushi Kiyoki, “An Interconnection Method for Heterogeneous Knowledge Bases by Utilizing Wikipedia”, 20th Technical Committee on Semantic Web and Ontology, http://sigswo.org/papers/SIG-SWO-A803/SIG-SWO-A803-04.pdf, SIG-SWO-A803-04, Jan., 2009, (in Japanese).

[8] Masakatsu Mori, Ryoichi Ueda, Nobutoshi Sagawa, and Hiroko Sukeda, “KaaS (Knowledge as a Service) for Value-added Business Creation”, Hitachi Review ,vol. 91, no. 07 600-601, Jul., 2009, pp. 56-59, (in Japanese).

[9] Minako Toba, Yasuhide Mori, and Daisuke Tashiro,”Proposal of “Knowledge Recommender” the Business-knowledge Sharing System and its Application to Building-energy-management Service”, IPSJ Transactions vol.53 no.1, Jan., 2012 , pp.149-162, (in Japanese).

[10] Freshmanmoney.com, “Income/Resident Tax Calculator, Japan”, http://en.freshmanmoney.com/tax/form/calc_form.html, (accessed 2013-02-12).

[11] Real Estate Information in Hiroshima, “You can calculate your house loan”, http://home.gr.jp/athome/file/loankeisan.html, (in Japanese), (accessed 2013-02-12).

[12] Amazon, “AWS”, http://aws.amazon.com/, (accessed 2013-02-12).

[13] Google, “Developers”, https://developers.google.com/appengine/, (accessed 2013-02-12).

[14] Zoho, “Creator”, http://www.zoho.jp/creator/, (in Japanese), (accessed 2013-02-12).

[15] Steffen Kächele, Jörg Domaschka and Franz J. Hauck, “COSCA: an easy-to-use component-based PaaS cloud system for common applications”, In Proc. of the First International Workshop on Cloud Computing Platforms, CloudCP’11, New York, USA: ACM Press., pp.1–6.

[16] Microlab Co. Ltd., ”XCute”, http://www.microlab.jp/index.htm, (in japanese),(accessed 2013-05-15).

[17] NHK, “Gatten, diet chart”, http://www9.nhk.or.jp/gatten/dietclub/, (in Japanese), (accessed 2013-02-12).

[18] Raymond W. Southworth, and Samuel L. Deleeuw, “Digital computation and Numerical Methods” McGraw-hill Book company, New York [“Mathematics for computer II - Numerical analysis-“, KYORITSU SHUPPAN CO., LTD.,pp.276-294, October, 1975, Japan] (in Japanese)

[19] M. Reza Rahimi, Nalini Venkatasubramanian, Sharad Mehrotra, and Athanasios V. Vasilakos, “MAPCloud: Mobile Applications on an Elastic and Scalable 2-Tier Cloud Architecture.”, 2012 Utility and Cloud Computing, IEEE Fifth International Conference on, Nov., 2012, pp. 83–90.