Computer Science and Engineering 2012, 2(2): 1-8
DOI: 10.5923/j.computer.20120202.01

Method of Adding Telephone Interface to Web Service (Implication to the Self-Care Support System for Life-Style Diseases)

Masatoshi Kawarasaki[1,*], Kyoji Konishi[1], Makoto Ohara[2], Tetsuya Igarashi[2]

[1]Graduate School of Library, Information and Media Studies, University of Tsukuba, Tsukuba, 305-8550, Japan
[2]Tsukuba University Hospital, Tsukuba, 305-8575, Japan

Abstract Various web services are offered over the internet and forms the social infrastructure indispensable to our daily life. Many of healthcare systems developed so far are also offered as web service. However, user interface of web services that require web access makes obstacle to elderly or inexperienced users. As a mean to mitigate such digital divide, we developed a generic method of adding telephone interface to web service and implemented it to the self-care support system 'iSMBG' for life-style diseases. In our method, by making software PBX cooperate with web service, input/output to web service can be achieved by ordinary telephone device and users can talk with web service interactively. In parallel to implement telephone interface to iSMBG, we reorganized the architecture and data structure of the original iSMBG web system to be more flexible in adding new functions and data types. Evaluation experiment revealed that the proposed method endures practical use.

Keywords Computer Mediated Communication, Health Information Management, Software Architecture, User Interface

1. Introduction

In the past decades, various medical-care or healthcare support systems have been studied, developed and deployed. Digital patient file systems are already deployed in many hospitals and Electronic Health Record (EHR)[1] is gathering attentions to share patient data among medical agencies. Recent work explores adding wireless connectivity to medical devices for remote monitoring of patients' vital signs, thus improves timely treatment leading to a better health care system[2]. These systems are mainly physician centric. On the other hand, 'Wellness support' of NTT docomo[3] and WiiFit of Nintendo[4] collect health information from a patients (or a patient-to-be) and return a result intelligently. These systems are mainly patient centric. Recent works explore the use of expert system[5], mobile game[6] and policy based data management[7] towards innovative patient health care support system. It should be noted that all these systems are provided as 'web service'.

We developed a self-care support system for diabetes patients, named 'iSMBG'[8], and performed a demonstration experiment in a diabetes clinic to validate its effectiveness. Although the number of patients and patients-to-be of life style diseases such as diabetes is increasing, achievements of medical treatment are not effective enough. To improve this situation, we designed iSMBG to assist doctors and patients interactively in patient's self-care activities. iSMBG is a web-based system. At the time of blood glucose measurement, a patient enters his/her data to a cell phone to be sent and stored in i-SMBG server. The server analyzes patient data automatically and sends an alert mail if self-care control is not sufficient.

The validity of the system was confirmed with this experiment, but the problem was that the elderly patients or the patients who are inexperienced in handling information devices did not want to participate in the experiment. Actually, the web-interface of iSMBG was a bottleneck for elderly patients. To mitigate such digital divide, we developed a generic method of adding telephone interface to web service and implemented it to iSMBG.

Regarding the addition of telephone interface to web service, some methods are foreseen. The most likely method is remaking software PBX. There are some examples in e-commerce or in call centers that use IVR (Interactive Voice Response) and voice recognition[9]. But these are not general-purpose systems. More generic method would be the use of VoiceBrowser whose standards are being developed in W3C[10]. VoiceBrowser includes VoiceXML which allow XML describe voice-applications just like HTML describes visual applications. It also includes supplementary XML such as CCXML[11] which describes the

* Corresponding author:
mkawa@slis.tsukuba.ac.jp (Masatoshi Kawarasaki)

Published online at http://journal.sapub.org/computer

call control of the VoIP telephone service, and SSML[12] which describes the volume and pitch of voice. By using the function in this VoiceBrowser, it would be possible to add telephone interface to web service. But these standards are large in volume and available open software is very limited such as to a simple VoiceXML interpreter[13] or open source IVR[14]. In the current situation, it is difficult to cooperate with various web services by using VoiceBrowser.

So we developed a new method of adding telephone interface to various web services that is easy to implement and has high scalability. In our method, by making software PBX cooperate with web service, input/output to web service can be achieved by ordinary telephone device and users can talk with web service interactively. Furthermore, fax output of user's web page and the use of voice message become available.

In parallel to adding telephone interface to iSMBG, we reorganized the architecture and data structure of iSMBG. As the original iSMBG was elaborated only to achieve a limited set of functions and data types, it was not flexible enough as a web-system to add new functions or interfaces. In order to allow iSMBG to handle voice and/or image data other than text data, as well as to manage various users such as nurse and pharmacist in addition to patient and doctor based on their respective roles, we reorganized iSMBG by using latest technologies of web service. In the new iSMBG, we adopted Model/View/Controller (MVC) framework for web-service. We also introduced role function for right management and adopted jQuery for rich user interface. We added telephone interface to this reorganized iSMBG and carried out evaluation experiment. It revealed that the proposed method endures practical use.

The rest of this paper is organized as follows. In the first half of this paper, we describe about iSMBG and its reorganization. Section 2 describes the objectives and overview of iSMBG and Section 3 describes the architecture and data management method of new iSMBG with specific examples. The method of adding telephone interface to web service is described in the latter half of this paper. Section 4 proposes the generic method of adding telephone interface to web service and evaluates its performance. The result of demonstration experiment of voice-guide using new iSMBG added with telephone interface is also described. Finally, Section 5 concludes this paper.

2. Self-care Support System for Lifestyle Diseases (iSMBG)

2.1. Background and Overview

Many of the diabetics doing insulin treatment are performing self-care of diabetes through SMBG (Self Monitoring Blood Glucose) to confirm control status. This is a very effective way but, in many cases, the result of SMBG is not well reflected in self-care improvement. In fact, even the basic self-care such as weight measurement or blood pressure measurement, which seems to be very easy, is not performed enough.

On the other hand, it's difficult for medical doctors to read the blood glucose data recorded on a patient file during the last one or two months, and judge the control status of the patient correctly in limited medical examination time. Furthermore, as diabetes specialists are few to the number of patients, diabetes specialists need to rely on family doctors in near-by area.

We developed iSMBG as the system that can support both of patients and doctors with the purpose of improving the quality of medical examination and treatment of lifestyle diseases such as diabetes which is surely a national disease today, by using the cellular phone which is quite familiar living tool already.

The system configuration of iSMBG is shown in Figure 1. Figure 2 shows the data entry screen of a cellar phone and Figure 3 shows doctor's web page for patient management.

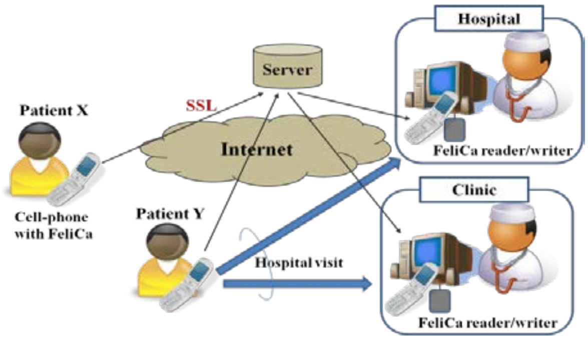


Figure 1. iSMBG system configuraion

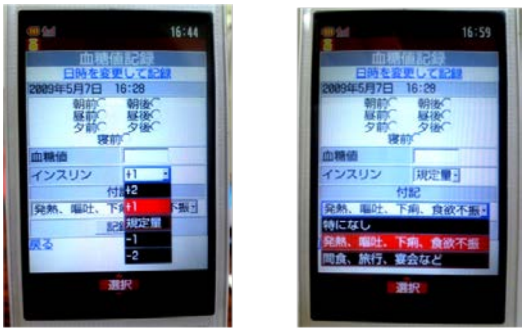
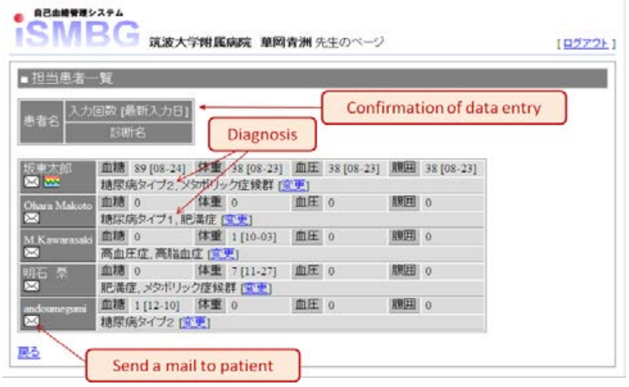


Figure 2. Input Screen of Cellar Phone


自己血糖管理システム iSMBG 筑波大学附属病院 華岡青洲 先生のページ [ログアウト]

■担当患者一覧

患者名	入力回数 [最新入力日] 診断名

坂東太郎	血糖	89 [08-24]	体重	38 [08-23]	血圧	38 [08-23]	腹囲	38 [08-23]
	糖尿病タイプ2, メタボリック症候群 [変更]							
Ohara Makoto	血糖	0	体重	0	血圧	0	腹囲	0
	糖尿病タイプ1, 肥満症 [変更]							
M.Kawarasaki	血糖	0	体重	1 [10-03]	血圧	0	腹囲	0
	高血圧症, 高脂血症 [変更]							
明石 泰	血糖	0	体重	7 [11-27]	血圧	0	腹囲	0
	肥満症, メタボリック症候群 [変更]							
andoumegumi	血糖	1 [12-10]	体重	0	血圧	0	腹囲	0
	糖尿病タイプ2 [変更]							

戻る

Figure 3. Doctor's web page for patient management

2.2. Main Features of iSMBG

There are three features in iSMBG. Firstly, the patient data is recorded in the server and analysed in advance of medical examination. Accordingly, a doctor can easily make diagnoses accurately in short time.

Secondly, it provides interactive self-care support. For example when hyperglycaemia continues or when coming to have continuous high blood pressure, warning mail is sent automatically from the system to user’s cell phone.

Thirdly, it achieves community medical care alliances. As patient data is put in the server, data can be shared among all medical agencies keeping appropriate security, thus it becomes easy to examine a disease and cooperate.

3. Rebuilding iSMBG

3.1. Design Principles

The demonstration experiment of iSMBG revealed that handling of web interface was difficult for many of elderly patients. To resolve this issue, we decided to add telephone interface to iSMBG. At the same time, we recognized the need of rebuilding iSMBG web-system architecture to facilitate adding new interfaces and services. Design principles of new iSMBG are described below.

3.1.1. More Flexible User Interface

In the original iSMBG, a data entry from a patient was accepted only through the web interface of a cellular phone. This was a bottleneck for many of iSMBG users who are aged patients. We decided to add the data entry interface for telephone voice to the web interface of a cell phone. At the same time, we intended to improve the flexibility in data types to allow optional forms as well as a text to include voice message from a telephone and image data from a cell phone. Example of image data could be a plate image for calorie calculation.

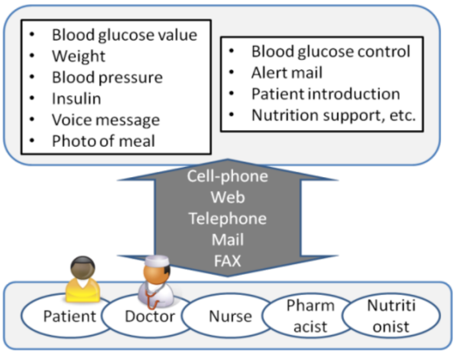


Figure 4. Cloud Service as a Bridge between Patient and Hospital

3.1.2. Enhanced Self-care Support

Original iSMBG was designed to promote the cooperation between patient and doctor. We expanded iSMBG users to include such as nurse, pharmacist, hospital staff and dietician, and made them available to participate in the patient support according to their respective roles.

With this expansion, a nurse can respond to voice message, and a dietician can analyse a plate image data sent from a patient and provide appropriate advice. Furthermore, family members of a patient can enter blood glucose data to iSMBG on behalf of the patient.

In this manner, the new iSMBG provides something like a cloud service that bridges between patients and hospitals as illustrated in Figure 4.

3.1.3. Enhanced Doctor Support

The original iSMBG supported a doctor through visualization of patient data and web page for patient management. We enhanced patient management capabilities for doctors in the new iSMBG. For example, we added the automatic calculation capability of the prescription amount of insulin by recording the category and amount of insulin that the patient has been administered. Furthermore, we simplified user interface of doctor's patient management web-page, so that more comfortable medical examination and patient support can be performed.

3.2. New iSMBG Architecture

To achieve the design principles described above, we adopted the latest technologies to rebuild the original iSMBG by using Model/View/Controller (MVC) architecture for web service, role function for right management, and jQuery for rich user interface. Details are described below.

3.2.1. Use of MVC Framework

iSMBG was rebuilt using the web application framework RubyOnRails of MVC architecture. This enabled easy addition of new functions and interfaces to iSMBG. This also enabled mounting in PaaS (Platform as a Service) that is increasing in use in recent years. PaaS which supports RubyOnRails exists a lot already. By mounting in PaaS, it's possible to make storage capacity and processing power more scalable than general VPS (Virtual Private Server) and/or rental server.

3.2.2. Addition of Telephone Interface

By using software PBX, access to iSMBG system was made possible from public telephone network. We developed the middleware which connects a main system with software PBX so that the script for telephone interface can be written in *View* of the main system. This makes it unnecessary to rewrite the original database input/output program for the telephone interface, thus facilitate the addition of telephone interface functions. Details are described in Section 4.

3.2.3. User Right Management by Role Function

Only patients and doctors were managed by the data structure of original iSMBG. So, every time when an additional user of iSMBG such as nurse or nutritionist is needed, it was necessary to add a new data structure. As a result, data structure of the whole system became complexly interrelated and the scalability was decreased

We changed the architecture so that all users of a main system are managed by the *users table* and their roles (attributes about user's rights) are managed by the *roles table* separately. The *roles users table* which manages the relationship between the role and the user was established, so that user's right is controlled by the role to which the user belongs. Details are described in Section 3.3.

The access control list (ACL) is established in *Model* of a main system and manages what kind of right each role has.

3.2.4. Rich User Interface Using jQuery

The doctor's patient management screen interface was made by tab structure like Figure 5 using the framework of JavaScript called jQuery.

When a doctor chooses a patient from the charged patient list, a tab of the chosen patient appears. When the doctor wants to make an operation to the patient, such as a calculation of the insulin prescription amount, he pushes a relevant button on the page. A dialogue box appears and the operation can be completed in this dialogue box. A doctor can browse patient information efficiently because he/she can wander back and forth between tabs with a small number of transitions.

The dynamic information shown on the web page is exchanged by JSON (JavaScript Object Notation) which uses Ajax (Asynchronous JavaScript + XML). As any resource can be accessed by REST (Representational State Transfer), third party applications that conform to iSMBG can be developed easily.

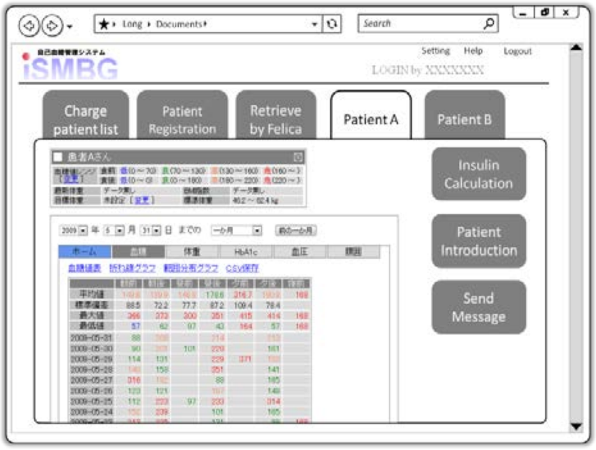


Figure 5. Interface mock-up of the patient management page

3.3. Data Management Method

3.3.1. Database Structure

Data management of the new iSMBG is shown in Figure 6, 7 and 8. Each box interconnected by lines represents a table, and a line represents a reference to a foreign key of another table. The colour represents the data type, where red indicates *String*, yellow indicates *Integer* and green indicates *Time*.

Data management system is structured so that tables are interconnected centering on *users table*. All the users who use this system are managed by *users table*. The *users table* includes only minimum information needed for user management, such as user name, login name and password. This is because the *users table* is very frequently accessed by login operations, it should keep a small size to ensure high performance.

Doctor information and patient information are not stored in *users table*, but stored separately in *doctors table* and *patients table*, respectively.

A *doctors table* stores such as medical agency information and a *patients table* stores such as upper and lower limit of blood glucose value, weight and height data of corresponding patient.

On the other hand, in order to describe the access right, the role function is managed by the *roles table*. The *roles table* includes users such as a system administrator, a doctor, a patient and a nurse, and the relationship with a particular user in the *users table* is managed by *roles_users table*. Relationship between a doctor and a patient is managed by the *charges table*. As a doctor and a patient are users, this relationship is a relation between one *users table* to another *users table*. (See Figure 7)

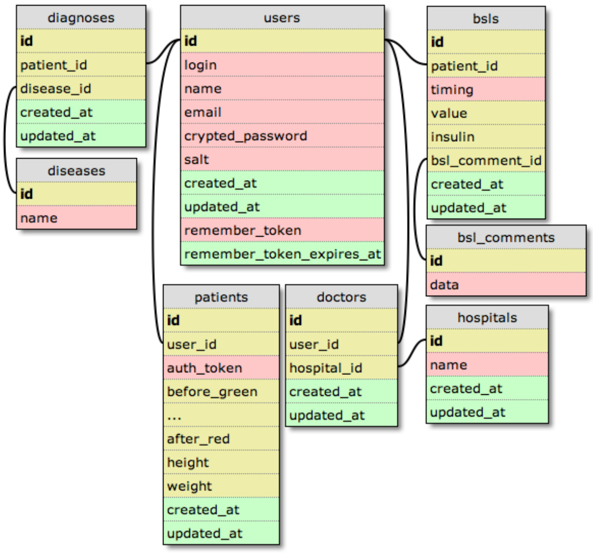


Figure 6. Database scheme for blood glucose management

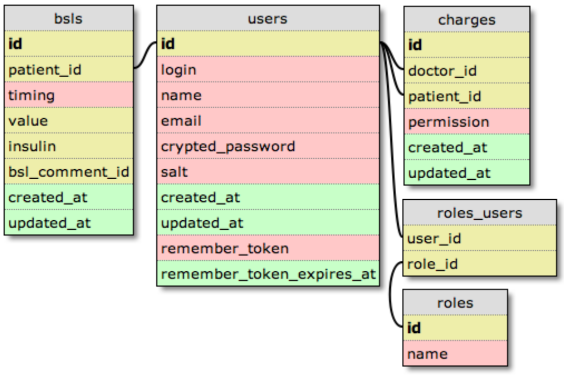


Figure 7. Deputy entry function and DB schema

Figure 6 illustrates the database schema for the blood glucose level management. Blood glucose level is managed by the *bsls table*, where *timing* means a timeframe such as before breakfast or after dinner, *value* means blood glucose value, and *insulin* means insulin information. Patient's annotation about blood glucose value is managed by the *bsl_comments table*, and is connected by a foreign key within the *bsls table*. Patient's annotation is not limited to a text but could be a voice message. For this purpose, the *mimetype* is introduced in *data type* on the *bsl_comments table* to distinguish a data-type such as character string, voice and any other data format.

3.3.2. Functional Behavior

How the management system works is explained by specific examples in this section.

(a) Deputy Entry of Blood Glucose Value by Third Party

Deputy entry function enables a third party user to enter SMBG data to the system on behalf of a patient. Figure 7 shows a database scheme that realizes a deputy entry function. Let us suppose that A is charged of deputy entry of B's SMBG data. Firstly, the system reads the record of deputy A in the *users table* as well as the relationship between *roles_users table* and *roles table* to examine whether A is surely a deputy of patient B. Then, the system reads a charge client list from the *charges table*. If A is confirmed to be a deputy of B, the system enables A to enter Patient B's SMBG data to the system. Then, A enters B's SMBG data in *bsls table*, thus completes the deputy entry operation. Deputy entry cannot be achieved when A is not registered as a deputy entry person in the *roles_users table*, nor when B is not registered to the system.

(b) Referral

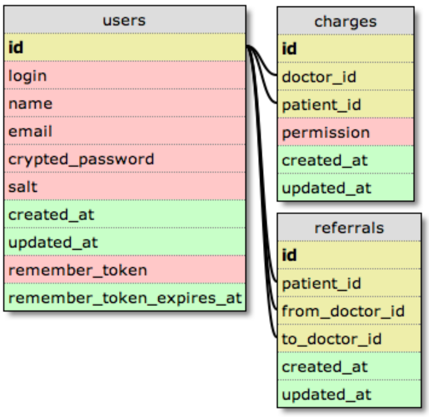


Figure 8. Referral Function and DB schema

Figure 8 illustrates the database scheme that realizes the referral function. Let us suppose that doctor *A* refers patient *C* to doctor *B*. *Referrals table* stores 'from *A* to *B*' relationship for each referral. *Charges table* manages the relationship between a doctor and a patient. The relationship is renewed after referral is completed. Thus the *referrals table* is used only for a stock of the temporary information when a referral is made.

The referral function operates as follows. When doctor *A* refers his/her charge patient *C* to doctor *B*, a record is generated on the *referrals table*, that includes respective *id* of the charge patient, namely doctor *A* and doctor *B* on the *users table*. A system reads relevant records on the *referrals table* automatically and eliminates the records of doctor *A* and patient *C* from a *charges table* at the time of the first medical examination by doctor *B*, and adds a charge patient *C* to doctor *B*. A referral is completed by this procedure.

It's possible to maintain the right in the *charges table* so that doctor *A* also reads patient *C*'s information after the referral to doctor *B* is completed. It's also possible to change doctor *A*'s right after referral, by modifying permission in the *charges table* regarding possible operations and available patient data.

4. Adding Telephone Interface to Web Service

Adding telephone interface to a web service is to connect PSTN (Public Switched Telephone Network) users to a web server via VoIP providers by using software PBX (Private Branch eXchange) and to make the web service available to PSTN users through the telephone interface as shown in Figure 9. A patient can enter health data and receive response to and from the web service through telephone interface by operating the telephone dial according to voice guide, thus allows the patient talk with his/her doctor interactively. Fax and voice message can also be utilized effectively. For example, a patient can receive a graph of blood glucose value transition by fax, and can deliver his concern about self-care to his doctor through a voice message.

In order to add a telephone interface to a web service, it's necessary to share the database (in the following, DB) between web service and telephone service, and make database contents deliverable to the users through respective interfaces.

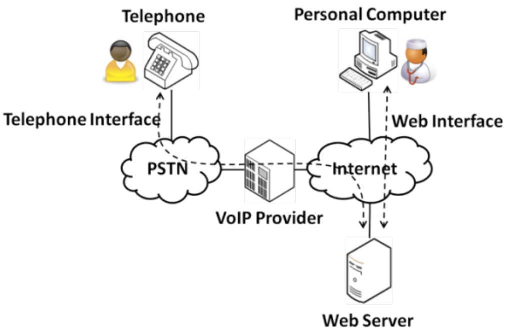


Figure 9. Addition of telephone interface to web service

Figure 10 shows the comparison of existing and proposed methods. In the existing method, access to web service and to common database is achieved by an external program of the software PBX. In the proposed method, by adding HTTP input/output function to software PBX, input to and output from database is performed only by the web service.

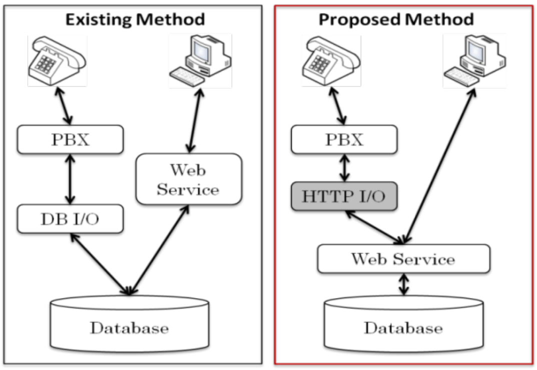


Figure 10. Comparison of existing and proposed method

4.1. Method of Adding Telephone Interface

A lot of software PBX has been developed with the spread of VoIP, and most of them have IVR function. The IVR function is used in enterprise call centers, where the voice file prepared beforehand is played on the user side to perform auto responding by voice according to the user's dial control. To add telephone interface to web service using IVR functions offered by software PBX, it's necessary to develop software for respective web services. This makes it difficult to offer various web services with telephone interface.

In this section, method of adding the telephone interface to web service is proposed. Proposed method is applicable to various web services in general, not limiting to iSMBG.

4.1.1. Existing Method and its Problem

To provide IVR service, various conditions are usually described using the functions attached to software PBX. Figure 11 shows an example of Asterisk that is a typical example of software PBX. In Asterisk, it's possible to describe the response conditions of IVR in a procedural way by using AEL (Asterisk Extension Language). Furthermore, it's possible to write the user's entry data in the database because the external process can be performed by the special form as AGI (Asterisk Gateway Interface) from AEL. Similarly, it's possible to read data from the database and hand the result to telephone users.

By operating web service and common database in this way, it's possible to read out the contents offered by web service using the telephone interface. However, as IVR functions as attached to the software PBX are limited, it's difficult to provide complicated services by using this mechanism.

On the other hand, the necessity to prepare similar programs both in the web service side and the PBX side comes out because the intersection with the web service is only the database. As a result, it's necessary to mount a large-scale program both in the web side and the PBX side in order to provide PSTN users with interactive responses or user specific recommendations from the web service.

Furthermore, a developer has to describe the configurations of IVR in detail using different languages specific to each software PBX, thus requires a big effort to the developer who tries to add telephone interface to web service that requests entry of various contents.

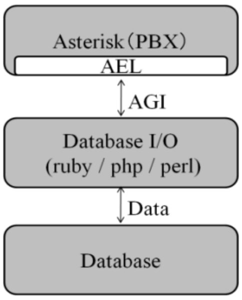


Figure 11. Example of an image with acceptable resolution

4.1.2. Proposed Method

(a) Design Principles

Proposed method includes two design principles. Firstly, the IVR function of software PBX is extended so that it can access to the established web server, play a voice file according to acquired XML, and transmit the user's entry data to web server. Extended function, as denoted by HTTP I/O in Figure 10, behaves like a web browser. It reads XML data from the web server and transmits a voice file, user input format and input timeout period to the PBX. When PBX receives user's entry, it delivers the input data to the callback URL through HTTP I/O.

Secondly, a web service is assumed to have the Model-View-Controller (MVC) architecture which is adopted in many of web frameworks. MVC architecture consists of Model which processes data, View which deals with display and data output, and Controller which controls View. We aimed at the architecture that enables to provide the users with the result from web service through the telephone interface only by just making alterations in View for HTTP I/O on the PBX side, keeping Model intact where various processing is performed.

(b) System Behaviour

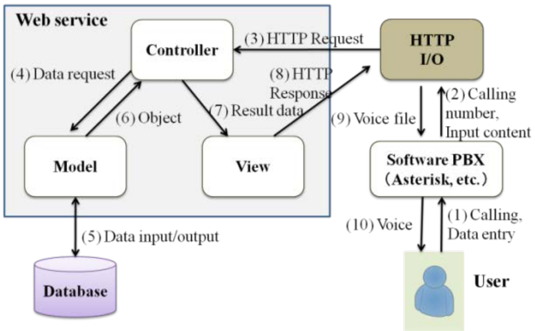


Figure 12. System behavior

The overall system behaviour is shown in Figure 12.

(1) A user makes a call to PBX and sends input data. (2) PBX sends a caller ID and the input contents to HTTP I/O. (3) HTTP I/O sends a HTTP request with a caller ID and the input contents to Controller on the web service side. (4) Controller requests Model for data acquisition from DB. (5)Model performs read and write from DB according to the request from Controller. (6) Model receives a result from

DB and returns the object to which O/R mapping was made to Controller. (7) The object which was returned from Model is processed at Controller and handed to View. (8) Processing result from Controller and XML generated from an output template are returned to HTTP I/O. (9) HTTP I/O issues a directive of voice file replaying to PBX according to XML description. (10) Voice is played by telephone and data entry is suggested to the user.

By repeating this procedure, all the services currently provided by web service can be provided through telephone interface.

4.2. Performance Evaluation

In the proposed method, when compared with existing method, voice file downloading on the web server side and real-time analysis of XML by HTTP I/O become overhead and may influence the performance in response time. Performance degradation caused by this overhead of the proposed method is evaluated in this section.

4.2.1. Evaluation Model and Conditions

To measure the overhead time in voice file downloading, we established an evaluation system where a web server and a PBX server are connected by 100Mbps line. Athlon64 2.2GHz was used as CPU for both servers, and CentOS5 was used for OS. RubyOnRails was used for web application framework and Asterisk was used for software PBX.

We measured the time length required to play voice after the user enters a key by telephone. Voice was compressed by 12.2kbps GSM-EFR voice coding system developed for cellular phones, and has the duration of 20 seconds.

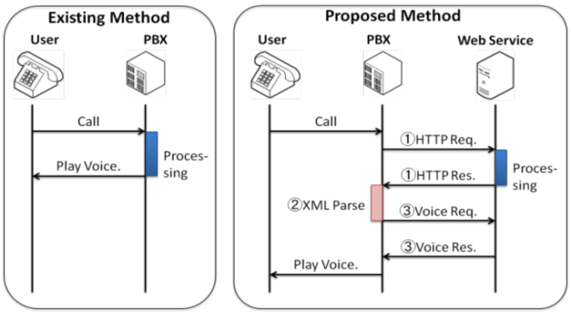


Figure 13. Signal sequence for performance evaluation

We examined whether the overhead time of the proposed method is tolerable in actual use under these conditions. Following three kinds of overhead time as shown in Figure 13 were measured.

(1) Time length between a HTTP request and its response.
(2) Time length required for XML analysis (XML Parse)
(3) Download time of the voice file

It should be noted that required time to process input contents and get a result is not included in above three. This is because a similar overhead is needed as well in the existing method.

4.2.2. Evaluation Results

The measurement results of overhead time are shown in Table 1. We measured these metrics 20 times for each overhead and took an average. The result revealed that 80% of overhead time is occupied by download time of the voice file. Although the time required for HTTP request/response and XML analysis don't fluctuate a lot, download time of voice guide is influenced a lot by the voice file size (the length of the voice guide). There is almost no overhead time on the PBX side because the processing time of HTTP I/O is simple and the voice file is compressed in low bit-rate.

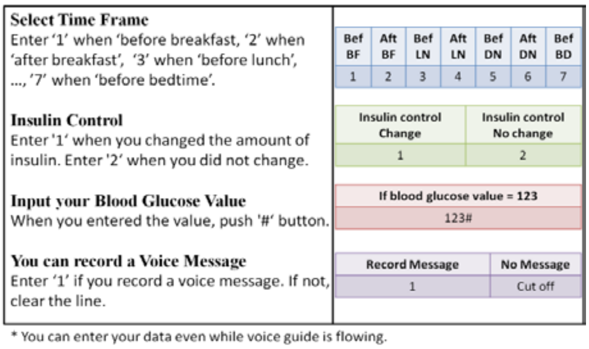
From this evaluation result, extension in processing time caused by the overhead was approximately 0.14 seconds. It revealed that overhead time is sufficiently short and hardly sensed by the user.

Table 1. Results of overhead time

From HTTP request to its response	23msec
XML parse	3ms
Download	113ms

4.3. Evaluation Experiment Using iSMBG

We implemented above mentioned telephone interface to iSMBG and evaluated the usability of voice guide through a demonstration experiment. Ten students of Tsukuba University participated in this experiment as monitors. Each monitor student enters the virtual data to iSMBG through the telephone interface according to the voice guide manual as shown in Figure 14. The monitors enter five input items (four items of Figure 14 and one item for entry confirmation) twice a day during one week,



Select Time Frame
Enter '1' when 'before breakfast, '2' when 'after breakfast', '3' when 'before lunch', ..., '7' when 'before bedtime'.

Bef BF	Aft BF	Bef LN	Aft LN	Bef DN	Aft DN	Bef BD
1	2	3	4	5	6	7

Insulin Control
Enter '1' when you changed the amount of insulin. Enter '2' when you did not change.

Insulin control Change	Insulin control No change
1	2

Input your Blood Glucose Value
When you entered the value, push '#' button.

If blood glucose value = 123
123#

You can record a Voice Message
Enter '1' if you record a voice message. If not, clear the line.

Record Message	No Message
1	Cut off

* You can enter your data even while voice guide is flowing.

Figure 14. Voice Guide Manual

4.3.1. Evaluation Parameters and Experiment Results

The original purpose of the experiment was to investigate how the users acquire fluency in the telephone interface. However, on the way of investigation, we noticed that it give us useful design parameter values for user-machine interaction of the proposed telephone interface. These are three parameters described below with respective experimental results.

(a) Appropriate response time from data entry

Response time is the time until the service returns a result by voice guide after user's data entry. When it's too short, the user may miss hearing the first part of voice guide, and when it's too long, the user has to wait for the result. From the result of questionnaire, 80% answered that one second was fine, and 20% preferred a slower response.

(b) Upper limit in the number of entry items per one telephone communication

In this experiment, five entry items were requested during each telephone call. From the result of questionnaire, 90% answered that five input items were appropriate, and 10% answered that it could be a little bit more.

(c) Upper limit of acceptable time length of a telephone call required to answer all the entry items

This is the acceptable time until the user disconnects the telephone after he/she makes a telephone call to the service. From the result of questionnaire, 30% answered that 60 seconds was the acceptable upper limit and 70% answered that 120 seconds was the limit. The average was 102 seconds.

4.3.2. Analysis of Experiment Results

It revealed that an audio response from a system needs to be sent out at least more than one second from user's data entry. It means that the total overhead time of 0.14 seconds as required in the proposed telephone interface is almost negligible level.

As for the length of voice guide which mainly determines the total overhead time, we calculated the upper-limit time needed to each entry item by the following formula;

(Total time length) / (Number of entry items)
= (Time length of each entry item).

The result was less than 21 seconds. This could be interpreted that 20 seconds of voice guide, as used in the experiment, is sufficiently long and appropriate considering the time required for data entry and additional messages that flow besides the entry item.

5. Conclusions

We proposed a method of adding telephone interface to web service as a measure to mitigate digital divide for elderly or inexperienced people who are not familiar to web interface. By expanding the IVR functions of software PBX and enabling coexistence with MVC architecture, the proposed method reduces workload in development and facilitates adding new functions. The result of evaluation experiment shows that increase in overhead is little and the response time fitted into a practical area.

Although we used proprietary XML schema for the communication between web server and PBX server, the use of standardized schema of VoiceXML or CCXML would be better to make it a more general-purpose system. This is a subject for further study

As for reorganization of iSMBG, it is the problem within a web system and has no relation to general-purpose property of the proposed method of adding telephone interface to web service. Proposed reorganization improved the flexibility of iSMBG as a web-system in adding new functions and/or data types.

As for demonstration experiment, we only carried out the evaluation of voice guide usability through telephone interface. We plan to carry out the demonstration experiment in a diabetes clinic again to examine whether the new iSMBG added by telephone interface is favorably accepted by elderly patients who are not familiar with web-interface, as well as to examine the usefulness of deputy entry service or referral service that we have introduced in new iSMBG

REFERENCES

[1] Marco Eichelberg, Thomas Aden, Jörg Riesmeier, Asuman Dogac, and Gokce B. Laleci, A survey and analysis of Electronic Healthcare Record standards. ACM Comput. Surv. 37, 4, pp.277-315, December 2005.

[2] W. H. Maisel. Safety issues involving medical devices: Implications of recent implantable cardioverter-defibrillator malfunctions. Journal of the American Medical Association, 2005

[3] NTT docomo Wellness-Support - http://www.docomo.biz /html/solution/all/wellness/, 2011

[4] Wii Fit homepage. http://wiifit.com, 2011.

[5] Gang Luo, Chunqiang Tang, and Selena B. Thomas. Intelligent personal health record: experience and open issues. In Proceedings of the 1st ACM International Health Informatics Symposium (IHI '10), Tiffany Veinot (Ed.), ACM, New York, NY, USA, pp.326-335, 2010

[6] J. Sunwoo, W Yuen, C Lutteroth and B. Wünsche, Mobile games for elderly healthcare, Proceedings of *CHINZ'10,* Auckland, New Zealand, pp.73-76, 2010

[7] Min Mun, Shuai Hao, Nilesh Mishra, Katie Shilton, Jeff Burke, Deborah Estrin, Mark Hansen, and Ramesh Govindan. Personal data vaults: a locus of control for personal data streams. In Proceedings of the 6th International Conference (Co-NEXT '10), 2010

[8] M. Kawarasaki, T. Asano, M. Ohara, T. Igarashi, Development and validation of cell-phone based interactive self-care support system for life-style diseases, Japan Journal of Medical Informatics, Vol.29, No.5, pp.201-210, 2010 (in Japanese)

[9] WIPO Patent Application WO/2001/071543, System and method for using voice over a telephone to access process and carry out transactions over the internet, Quack, Com, Sep. 2001

[10] W3C "Voice Browser" Working Group - http://www.w3.org /Voice/, 2011

[11] D. Amyot and R. Simoes, Combining VoiceXML with CCXML, IEEE Proc. of Consumer Communications & Networking,(CCNC07), 2007

[12] O. Kiselyov, SXML Specification, ACM SIGPLAN Notices, 2002

[13] JVoiceXML (Open Source Voice XML Interperter), http:/ /jvoicexml.sourceforge.net

[14] Zanzibar OpenIVR project, http://www.spokentech.org