I-CAN Disability Support System at the Royal Rehab Centre in Ryde

Noppachai Wongsai
Department of Computing
Macquarie University
Sydney, Australia
noppachai_wongsai@yahoo.com

Abstract

The Instrument for the Classification and Assessment of Support Needs, I-CAN, is a web-based tool for an assessment of support needs for people with disabilities. The assessment has been used to determine the current and future health care planning. The main tagger user of the existing system is support services. The development of the new version of I-CAN with additional functionalities is planned in the near future. In this paper, we develop the prototype of the web application for demonstrating how new web technologies can be implemented to achieve the requirements in the new version of I-CAN. An approach is based on applying new web technologies, such as AJAX and JavaScript libraries. This prototype, with a rich-attractive user interface application, can be further used and extended when other requirements are needed to be implemented.

1 Introduction

The World Health Organization (WHO, 2001) has established the conceptual framework of the International Classification of Functioning, Disability and Health (ICF) as the international standard for measuring health and disability for individuals and populations. The measures are based on individual body functions and structures, society activity and participation, and environmental factors. The ICF has been used in many countries, including Australia. The Royal Rehabilitation Centre, Sydney (RRCS), the University of Sydney and Industry Partners, and the Centre for Disability Studies (CDS) are the three organizations jointly responsible for developing an assessment instrument based on the ICF framework. It is called the Instrument for the Classification and Assessment of Support Needs, I-CAN, a tool to help physicians to determine the support needs for people with disabilities.

The first I-CAN version was paper-based. In late 2008, the latest I-CAN version 4.2 has been initially developed as a web-based system with HTML, PHP, and MySQL. It is available online at http://www.i-can.org.au/. This version is not only provided for physicians but also for support services. The instrument helps the support services to generate assessments identifying the areas in which supports are needed for the current and/or future health plans for disabled people. The support services can purchase an access to the system and do the assessment online. They can fill in personal data of the person with disability into the system directly during the interview process. In cases where an Internet access is not available, a paper form is used to collect data and those data are then entered into the system. Such data are used to generate the assessments for individuals.

The objective of the new I-CAN development is to improve the value and quality of the assessment by gathering both additional information from the person with a disability herself/himself and participation from their family and friends. With the availability of new web concepts and technologies, there is the potential to develop and implement these new requirements within the existing I-CAN system. In this paper, we present the overview of the existing I-CAN system, the developing approach and the implementation of a prototype that can support the above requirements, based on new web technologies and mov-

ing I-CAN towards becoming a Web 2.0 type application.

2 Existing I-CAN System

2.1 I-CAN Structure

Figure 1 shows the domain structure of data entry categories in the existing I-CAN version 4.2. Two major categories are person's health and well being, and her/his activities and participation in everyday life. Each category comprises several support needs that are measured by the frequency and level of support required. These support needs are balanced by personal goals, dreams, aspirations, current life situation, and supporting network. The structure is designed to support all disability types, at different ages and different levels of impairment.

Figure 1: Domain structure of the current version 4.2 (I-CAN 2008)

2.2 Architecture

The I-CAN system deploys the LAMP (Linux®, Apache, MySQL, and PHP/Perl/Python) architecture, well known as the foundation of many web applications. The system is simply developed in such a way that data access, business logic and user interface are separated into separate layers. The Data Access layer, in this case, is considered as a database where information about the person with disability is stored and retrieved. It is implemented with MySQL. The Business Logic layer (or application layer) normally contains Organizational Business Domain expertise. Many functions and mechanisms that are used in the assessment of support needs, and generating discharge reports, summarized reports and task lists are maintained in this layer. Most of them are written in the PHP scripting language. Finally, the User Interface (UI) layer (or presentation layer) takes responsibility for client presentation. The web presentation follows the UI framework that encourages implementing font end validations, and used style sheets and templates. It uses the events and data to invoke the appropriate business logic and specific logic, such as selectively showing or hiding fields.

The core functionalities of the I-CAN web application are segregated into three separate areas; (1) data entry and data assessment functions, (2) organisation function and (3) user account function. Data entry and data assessment function are used for collecting data of the person with disability, then processing, followed by generating discharge reports or/and custom reports for particular disabled people in various document formats (pdf, doc, etc.). Organisation and user account functions are used for managing organisation data and access accounts granted by the CDS.

2.3 Limitations

The three-layer web development model is the fundamental application architecture of the current I-CAN system. However, there are limitations. At the UI layer, plain static HTML pages in collaboration with JavaScript and CSS style sheets are used to represent the application presentation. The interface design for data entry was simply base on the structure used in the standardised paper form of the assessment which contains many sections of requested data (for example, there are 17 sections for data entry in Enter New Person function). This forces the logic of data entry to be separated into many web pages. Thus, when requested data are filled in and submitted, the application has to reload web page for displaying another section. This will have an effect on the application performance in terms of the data transmission time and bandwidth consumption. Furthermore, there is evidence that the logic and the application state are inappropriately managed. For example, one data entry section must be fully completed before the user can move to the next section; in the situation that a user cannot obtain the requested data for some reason he/she will not be allowed to continue. Similarly, at the Data Access layer, the database was simply designed according to the structure of the paper form. The tables in the database, as well as data fields and data definition, were constructed from each support need in the data entry

category (Figure 1) resulting in duplication of data content. It appears the database was designed disregarding normalization principles, ignoring referential integrity, encouraging one table to hold all domain values, using inappropriate keys and without using SQL facilities and stored procedures to maintain data integrity and data security.

3 Developing Prototype

The new version of I-CAN proposes an extension of the existing structure with increased functionalities and features. The new system is expected to handle a greater number of users, provide a familiar user interface, offer improved performance, and maintain a certain minimum level of reliability, availability and security. Figure 2 illustrates the new structure for the new version of I-CAN. Several new categories are introduced in order to collect specific information and allow participation between the person with disability and their family and friends. In this paper, two categories, Contacts and Personal Details features, are selected in developing of the prototype. How the existing system can be improved, and how the limitations of the current web application can be reduced are demonstrated. Data required in these two features will be entered into the system and will be managed by a user with authorised access to those data.

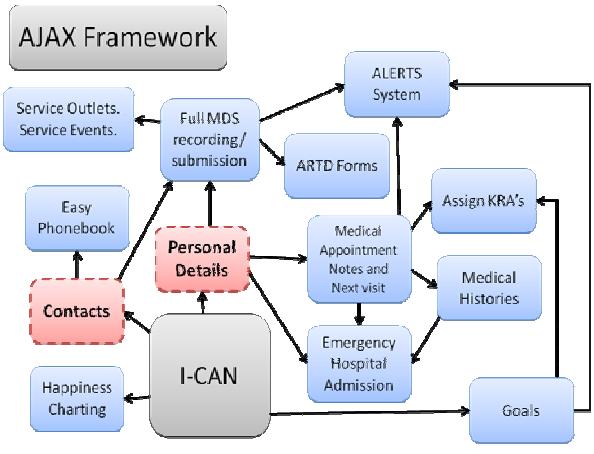

Figure 2: Extensional structure of the new version of I-CAN

The proposed architecture is still based on an open standard. The open-source LAMP Web platform is deployed as the most cost effective and comprehensive solution architecture. The proposed five layers concept of web development (Figure 3) is adopted as the application architecture. The three separated layers used in the existing system have been replaced by the concept of five levels of web development. This allows the individual modules with their own technologies and methodologies to be easily maintained with little or no impact on the other modules, hence improving accessibility and maintainability.

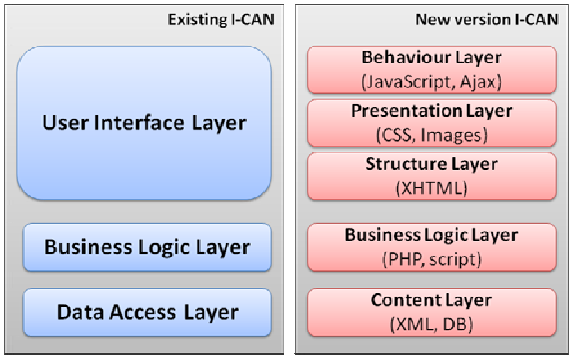

Figure 3: Three-layer and five-layer concepts

The five layers are the following:

- **The behaviour layer**: This layer defines how different elements behave using JavaScript and/or ActionScript that executes on the client site (or browser), and makes the user experience as easy as possible.
- **The presentation layer**: This layer presents the appearance of the web application and adds a visual design to the rendered output on the client site using images and CSS.
- **The structure layer**: This layer often involves the (X)HTML markup. However, it could also involve PDFs, SWFs, plain text, and even RSS, XML, AJAX or JSON.
- **The business logic layer**: This layer generally involves a server-side processing on a backend server in various environments that decides with task to perform according to requested information from the client and determines the returned information. Generally, this layer will interact with the data in the content layer.
- **The content layer**: This layer sits on the server and contains plain text in an XML format file and/or stored in a relational

database, also containing images, PDFs, SWF files and multimedia contents.

The UI layer is divided into three sub layers; behaviour, presentation and structure layer (Figure 3). The content layer is fully separated from the structure layer. To simplify it, *content* is stored in XML (or the database of choice), presented in (X)HTML (or PDF or Flash) and styled with CSS if it is presented with HTML. Moreover, data, structure and design are all more effective in conveying the contents. Nonetheless, added behavioral functionality can be powerful if executed properly on top of the other layers.

The implementation of the prototype is described in the following sections.

3.1 User Interface Design

The design of the UI concerns the presentation and structure layers. For designing the UI, the principles of UI design guiding Web 2.0 have influenced the main UI design decisions for this prototype. Fundamentally, the new generation of the web, known as Web 2.0, focuses on a user interface design that is easy to use and understand. The system should be as easy and intuitive as possible to use. Therefore, the pages should be consistent in look and feel, which helps the user in becoming familiar with the system faster. Furthermore, the pages and functions should be logically grouped to facilitate navigation between the pages. Functions should be placed where the user would most likely expect them without having to consult a manual first, and website design should encourage easy interaction by web surfers.

The design utilises the web browser as a means to present an easy to read web interface with different fonts and styles using CSS, and also to offer a pleasurable online experience with more features. New features such as Tab, Web Navigation, Round Corners and Light boxes are introduced in this UI design. For example, the Tab feature has been used to overcome the limitations of screen real estate and to fit better with the standardised paper forms of the health assessment. The Tab feature allows many sections of a data entry form to be included in a single web page and thus avoids breaking the logic across separate web pages. Also considered in design is cross-browser compatibility (Compatible with IE, Mozilla and Safari) and W3C Compliance which is achieved by following the W3C standard.

3.2 Database Design

The database design involves the content layer in the proposed five layers model. The database schema of the existing I-CAN web application, like with the UI design, was mostly designed according to the paper forms. There are, however, several drawbacks resulting in repetition and loss of information, inability to present certain information, integrity and maintainability issues. In this prototype development, database normalisation process has been used to redesign the database. The benefits not only enhance the overall database organization, but also reduce data redundancy, increase data consistency and better handle database security (Connolly and Begg, 2009). The major benefit, moreover, is to gain a more flexible database design which will support growing database and thus increase scailability.

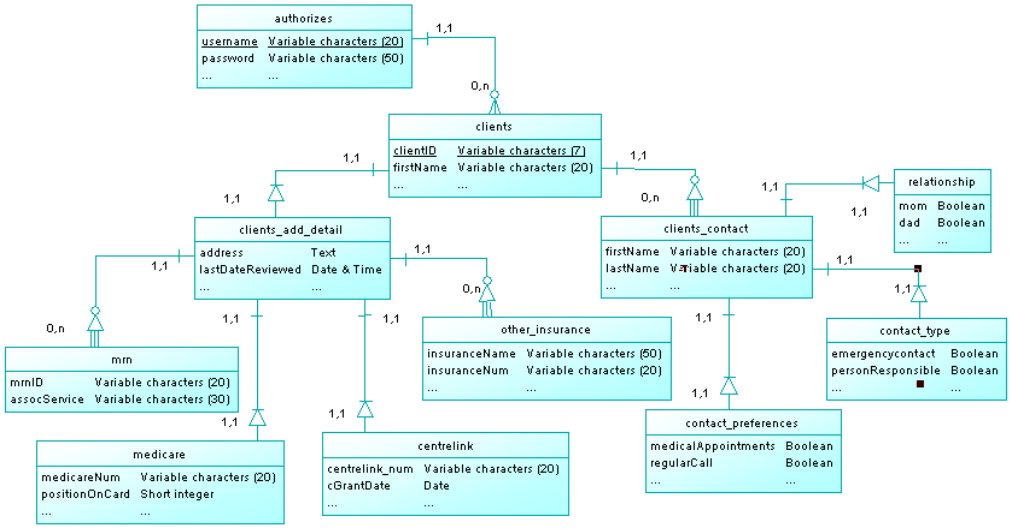

Figure 4: Conceptual Data Model of new two features

The database of the two features that are implemented in this prototype has been modeled using Sybase PowerDesigner tool as this tool offers database modeling from conceptual to physical data modeling and automatically generate the database schema for the desired physical database target (Sybase, 2009).

Figure 4 shows the conceptual data model of the two new features. The *client_add_detail* table is used to contain additional personal data of person with disability. It has one-to-one mandatory relationship with *client* table which already existed in the I-CAN relational database. It is decomposed into four tables, namely *mrn*, *medicare*, *centrelink* and *other_insurance*. The table *mrn* and *other_insurance* have one-to-many dependent relationship with *client_add_detail* table. The other tables, *medicare* and *centrelink* have a one-to-one dependent relationship with mandatory association. In contrast, the *clients_contact* table has one-to-many and dependant relationship with the *client* table. It is used to hold a number of alternative contact details of the person with disability and decomposed into three tables, namely relationship, *contact_type* and *contact_preferences*. They all have a one-to-one dependent relationship with the *clients_contact* table.

3.3 Client Site Scripting

The implementation of the prototype at the client site involves the behaviour layer. The comprehensive JavaScript library, called jQuery, is used. This fast and concise library simplifies HTML document traversing, animating, event handling, and AJAX interactions for rapid web development. The three main characteristics of jQuery are lightweight footprint, CSS3 compliant with supports CSS 1-3 selectors and cross-browser (IE 6.0+, FF 2+, Safari 3.0+, Opera 9.0+) (jQuery, 2009). In addition, it allows and facilitates to cooperate with other libraries, and comes with UI library (jQuery UI) that provides many widgets, interaction modules and themes. This UI library is used in the UI design and implementation. jQuery is a completely different way of writing JavaScript which is shorter and easier way. AJAX is another new web technology that has been applied in the behavior layer using jQuery (example code shows in figure 5). AJAX is a method of employing JavaScript, DHTML, and the XMLHttp behavior in the browser to provide truly dynamic content on a web page without the page-refresh. The XMLHttpRequest and ActiveXObject object, known as main mechanisms of Ajax, are encapsulated inside the jQuery library. This makes it easier to develop web application with rich interactive UIs.

```
$(document).ready(function(){
  // Ajax login mechanism using method post in jQuery
  $("#login_form").submit(function(){
    $.post("./php_backend/ajax_login.php",{
        user_name:$('#username').val(),
        password:$('#password').val()
        }, function(cbmeg){ //callback function });
     return false;
  });
});
```

Figure 5: Ajax script in jQuery

The main reasons for introducing AJAX technology are to improve the time waiting for data to be transmitted and completing a particular task, and the bandwidth consumption for the entire task (Charland, 2009). For example, in a login scenario, only the requested username and password string (approximately 100 bytes) are posted to the server and only a 1 byte boolean value is returned. Thus, a maximum 101 bytes of data (or parameters) is posted and received. If the same functionality were implemented using ASP.NET, in a conservative case, all the state data would be posted and the entire page (approximately 20 Kbytes) returned. Using AJAX would effectively produce 99.5% less traffic.

3.4 Server Site Scripting

The server site scripting occurs in the business logic layer where PHP script has been implemented. In the prototype, PHP pages are coded in classes. Object Oriented Programming in PHP is used to declare classes and using only objects and classes for connecting the database and manipulating the content with in the database. All the business logics that involve sending and retrieving data from the content layer are coded in OOP style. This addresses the limitations associated with the implementation of functionalities and business logics in traditional procedural style, since procedure code can be very messy and difficult to adapt later on. OOP code is easy to maintain, easy to understand and, importantly, easy to reuse and extend.

Another implementation in the business logic layer is AJAX-PHP script. These PHP pages are used as back-end scripting for AJAX functions. Typically, AJAX scripting on client site posts data or parameter to PHP page located on the server site, then PHP code will perform specific task depend on incoming data or parameter. It

could be querying, modifying content in the database and/or returning content back to the client site in the format of text or XML. Figure 6 shows an example of XML document containing selected contact detail of person with disability that return to AJAX script on client site. The returned document or content will be handled by callback function in AJAX script as shown in figure 5.

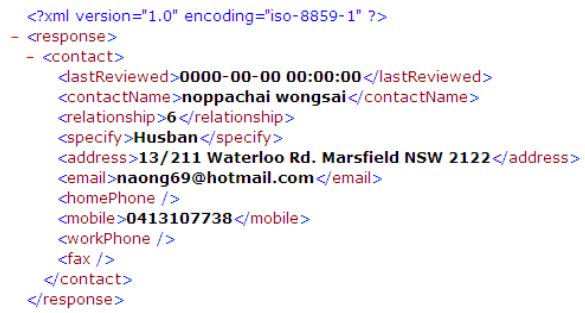

Figure 6: An example of Contact Detail information in XML document

3.5 Testing

Testing is an important part of each software development process, no matter which programming paradigm is used. It is the process to assure the quality of the developed products. Test Case scenarios are effectively planed and conducted for Web application testing (Nguyen, 2001). Test scenarios offer black box testing involving the execution of several test cases based on the user's activities or behaviors against the application. A check list of all test scenarios used in the prototype for testing are summarised in Table 1.

Table 1: Prototype testing scenario checklist

Web testing checklist	**Sub testing**
1) Functionality Testing	- All links checking
	- Forms checking
	- Cookies testing
	- Validate HTML/CSS
	- Database testing
2) Usability testing	- Navigation testing
	- Content checking
3) Interface testing	
4) Compatibility testing	- Browser compatibility
	- OS compatibility
	- Printing options
5) Performance testing	- Web load testing
	- Web stress testing
6) Security testing	

The web application and web site should be tested to ensure they meet the reliability, security, and performance goals necessary to effectively support business requirements. Increasing functionality and frequent changes in requirements present many testing issues in the web application environment.

4 Conclusion

The prototype demonstrated here shows that there is a possibility to extend the existing version of I-CAN to the updated version by employing the new web development concepts and web technologies, e.g. AJAX. The strategies and implementation approaches offer in this project can be used as a guideline for the further development of I-CAN.

References

Andre Charland. 2009. *Measuring the Benefits of Ajax*, Massive technology show Homepage, viewed 01/05/2009, <http://www.massivetechshow.com/resources/ajax.asp>.

Hung Q. Nguyen. 2001. *Testing Applications on the Web: Test Planning for Internet-Based Systems*, John Wiley & Sons, Inc.

I-CAN. 2009. I-CAN Homepage, viewed 21/04/2009, <http://www.i-can.org.au/>

jQuery. 2009. jQuery Homepage, viewed 27/04/2009, < http://jquery.com/>

Sybase. 2009. Sybase Homepage, PowerDesigner, viewed 24/04/2009, <http://www.sybase.com.au/products/modelingdevelopment/powerdesigner>.

Thomas M. Connolly and Carolyn E. Begg. 2009. *Database Systems: A practical Approach to Design, Implementation and Management*, 5th Edition, Addison Wesley Longman.

World Health Organization. 2001. *The International Classification of Functioning, Disability and Health (ICF)*, Geneva: Author.