Design of a Data Management Tool for the Diabetes Technology Center

T. Chhaya, K. Rosenthal and N.Yang

Abstract— **Diabetes, a disease preventing the body from breaking down sugars, affects over 20 million Americans each year. To conduct medical research and perform studies to enhance knowledge about the disease, the University of Virginia created the Diabetes Technology Center (DTC). This design project focuses on the creation and implementation of a database management system with a web-interface for the clinicians and researchers working on a study on Type I Diabetes. Our tool provides both a simple way to upload data into the database and easy extraction of data from the database for analysis. Other DTC studies rely on an outside technician to extract information from the database because the clinicians have little database knowledge. Our user-friendly tool allows these clinicians to extract data to analyze from the database.**

I. Introduction

THE University of Virginia's Diabetes and Hormone Center of Excellence founded the Diabetes Technology Center (DTC) in 1999 as part of an ongoing study examining the psychological and physical effects of diabetes. The DTC manages several studies, including IRB-HSSR #12252, with a small team of clinicians and representatives from the SIE department at the School of Engineering and Applied Sciences at the University of Virginia.

A. Problem Definition

For this project, the DTC is worked in conjunction with the Department of Systems and Information Engineering to create a database driven web-application to support IRB-HSSR #12252. This study is dedicated to the expansion of knowledge about Type I diabetes. The research focuses on the link between a patient's sensitivity to insulin, hypoglycemia, and a disorder known as Hypoglycemia Associated Autonomic Failure (HAAF).

Throughout the inpatient portion of the study, the medical assistants collect data concerning many medical attributes. Currently, the clinicians and researchers store all of their information in separate Microsoft Excel spreadsheets, making analysis difficult since there is no central data storage facility. The addition of our database and web-interface allows for central storage of data and simpler means for uploading and extracting data for analysis.

B. Study Background

Type I Diabetes is a condition, generally diagnosed during childhood, where the pancreas does not produce enough insulin – a hormone needed to process and break down sugars and starches into usable products. In order to break down these sugars, patients are prescribed a drug therapy of insulin injections. Hypoglycemia, or low blood glucose, may result if too much insulin is injected, not enough food is consumed, or if the patient is exceedingly sensitive to the prescribed insulin. Hypoglycemic Associated Autonomic Failure (HAAF) occurs when a person's hormonal response to counteract hypoglycemia lessens with each drop in blood glucose [1].

This design project focuses on IRB-HSSR #12252; a study on persons over 18 with Type I Diabetes. This study examines the possible link between insulin sensitivity, a person's risk for hypoglycemia and HAAF. They are examining two loops, shown in Figure 1. The first is that hypoglycemia increases insulin sensitivity, which then causes recurrent hypoglycemia. The second loop, also shown in Figure 1, is the theory that recurrent hypoglycemia leades to unawareness of the tell-tale signs of HAAF. This loop is studied by using hypoglycemic and euglycemic clamps to send the patient into hypoglycemia, and ultimately HAAF [2].

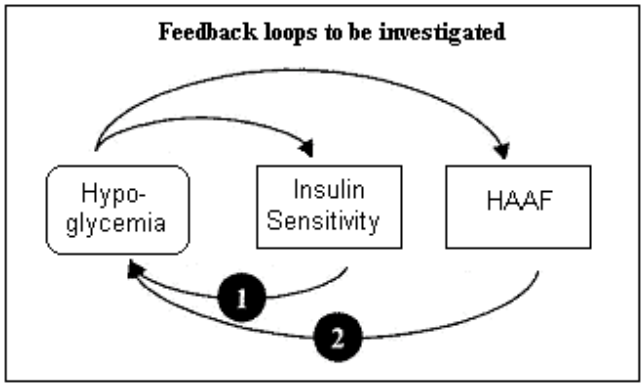

Fig. 1. Feedback loops investigated in study [2].

Manuscript received April 9, 2007. This work was supported in part by the Diabetes Technology Center at the University of Virginia and the Systems and Information Engineering Department of the School of Engineering and Applied Science.

Trisha P. Chhaya is a fourth year Systems Engineering student at the University of Virginia, Charlottesville, VA 22904 (email: tpc3d@virginia.edu).

Kara A. Rosenthal is a fourth year Systems Engineering student at the University of Virginia, Charlottesville, VA 22904 (email: kar4g@virginia.edu).

Ning Yang is a fourth year Systems Engineering student at the University of Virginia, Charlottesville, VA 22904 (email: ny3a@virginia.edu).

II. METHODOLOGY

The primary objective of the project was to simplify the work of the researchers and technicians in the study. Once the requirements were in place, we began the design, coding and testing of the uploading functionality of the tool. After this was complete, we built the analysis functionality of the tool which allows easy extraction of data from the database.

A. Problem Definition

Currently, clinicians record the data for each patient in a large Excel spreadsheet for the participants of the study. The DTC hopes to have a total of 140 patients by the end of 2007. If the current information storage system remains in place, the data will be convoluted and hard to analyze. Also, requests for data will overwhelm the researchers who currently manually load and extract data from the database. Due to the large projected increase in participants, the purpose for creating this system is to store participant data in one central electronic location. This database management tool will help not only to provide a deeper understanding of Type I diabetes, but will also simplify data access, decreasing the workload for technicians.

There were three distinct steps in the project: the design of the database, the upload functionality and the analysis tool. Some constraints that we faced in designing our system were that we had to use MySQL Control Center for the database and we were required to program using JSP (JavaServer Pages). This is due to the fact that we were adding our system to an existing system on the University of Virginia Medical Center's network which only supports JSP. JSP is a technology that combines Java code with static HTML and dynamically generated HTML [3].

B. *Database Design*

The first portion of the project was to design a relational database to hold the information collected during the study. The challenge in designing the database came in matching times and dates within the datasets. Since so much data was collected over the one or two days in the hospital, we needed a way for the data to be stored so all values for a specific time could be found.

There were two potential ways of solving this problem. We could either use a few tables with many columns or many tables with just a few columns in each. If we used just a few tables to store the data, it would be easier to code when we got to the upload and analysis tools because we would have fewer tables to dynamically join. By using several tables, we would be able to store all of the data without wasting storage space or having many null values because we would only have to store rows in each table where values existed.

The medical center's IDBM (internet database management) system had plenty of room on their servers, so we decided to use just a few tables with many columns. There are two main tables in our database which will be focused on for the remainder of the paper – the Clamp table and the Hormone table. Most of the data is taken during the hypoglycemic clamping portion of the study, which led to the creation of a Clamp table. This table would be populated by a spreadsheet designed specifically to match columns with the table. The Hormone data is also collected during the clamping portion of the study, but the values are not ready for long periods of time (up to three months). Because of this time difference, they needed to be in a separate table than the rest of the Clamp data. The primary keys for each table are SubjectID, Date and Time which allows joining of the tables on these three attributes.

C. Upload Functionality

Now that we had constructed the database for the project, we needed a front-end system to collect the data and add it to the database. To do this we designed and built a web interface to allow users to add data to our database. The assistants are moderate computer users and have little to no database or SQL knowledge, so the upload function had to be simple to learn and use. Also, there needed to be two parts to the upload tool; one for the Clamp table and another for the Hormone table. Since the columns in our Clamp table matched exactly with the columns in the spreadsheet, we had to write a program to load a comma-separated value (.csv) file directly into the database. The second part was to upload the Hormone data. This data would be stored in a different file for each hormone, so the assistants could load the data to the database as soon as it was ready. For the Clamp upload, we built a dropdown menu containing all of the SubjectID numbers that are participating in the study. This is filled dynamically, so any time a new participant joins the study, his or her SubjectID will be added to the dropdown. Below this is a text box with a browse button so the users can locate the appropriate file on their computer and then a submit button to load the file into the database. A screen shot is shown in Figure 2.

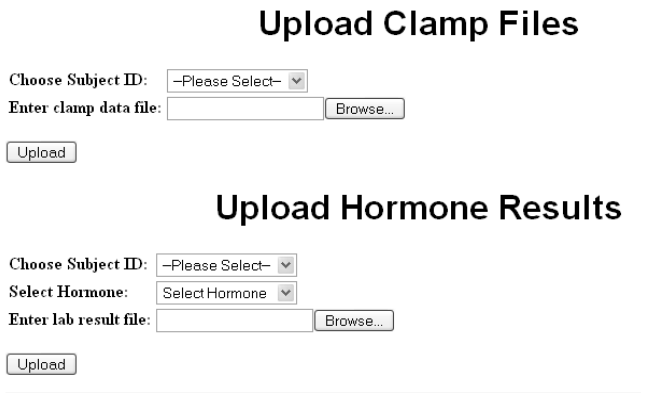

Fig. 2. Upload Functionality Screen Shot

The program performs several checks on the file to make sure that it is compatible with the database. It first checks to see if the SubjectID chosen in the dropdown matches the first column of the file being uploaded. If it does, it continues to check the file to be sure that it is a .csv file and that it has the appropriate number of columns. If these three things are true, the file is uploaded to the database. If not, an

appropriate error message is displayed and the user is prompted to try another file.

The second function of this page, the Hormones upload, was a bit more complicated than the Clamp upload. This is shown in the bottom half of Figure 2. For this part there is a dropdown to choose the SubjectID, another dropdown to choose the desired hormone, a browse button to select the file and a submit button to add the data to the database. This functionality also checks to be sure that the SubjectID in the dropdown matches the SubjectID in the first column of the file. It then checks that the Clamp file for the particular participant has already been loaded into the database. After, the program checks that the file has the appropriate number of columns and loads the file into the appropriate columns in the table according to the hormone chosen in the dropdown box. If it is the first file to be loaded into the Hormones table for that SubjectID, a row is inserted into the table. After the first file has been loaded, the subsequent loadings update the row in the table to add more data. If users try to add data for a hormone that is already populated in the database, they are prompted to choose a different hormone. This upload tool allows users to easily populate the Clamp and Hormones tables.

D. Analysis Tool

The next step of our project was to build an analysis tool so that the users could extract exactly the information they wanted from the database, again under the assumption that the users had no database or SQL knowledge. Our tool would not actually perform any analysis on the data, but would allow the users to pull the data onto the screen or into Excel, where they could find trends in the data. We needed to make a dynamic query builder so that the users could select exactly the variables that they wanted, build constraints, order the data by their selections, and output the file either on-screen or as an Excel-compatible file (.csv file). This tool also needed to be simple to learn and use.

We wanted to allow the users as much freedom as possible in creating the queries, while leaving the tool user-friendly. This tool was originally designed to be on one webpage so that the users could see all of the variables selected, constraints formed and orderings before submitting the query. But, in SQL, the results can only be ordered by variables that were selected. In order to account for this, the tool was split between two pages. One page, seen in Figure 3, has the variable selection, while the other page, seen in Figure 4, builds constraints and orders the results.

The variable selection page has all of the variables from which the users could choose. SubjectID, Date and DrawNumber are not included in this list because they are automatically included in every result set. After the users click on the 'Submit' button on this page, the tool directs them to the constraint selection page. This page has a series of steps that must be taken to build the query.

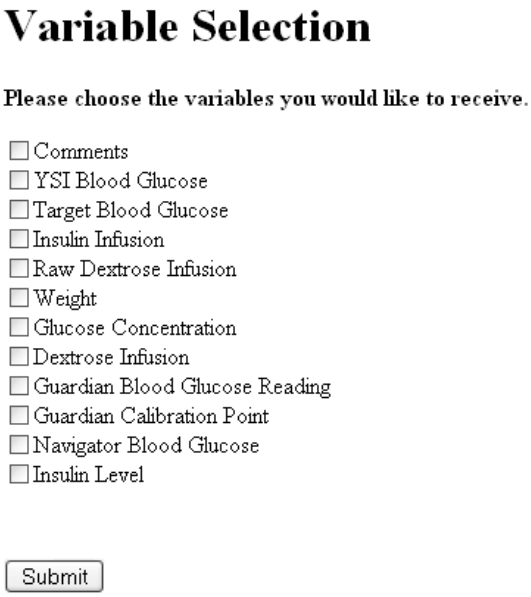

Fig. 3. Part of Variable Select Page.

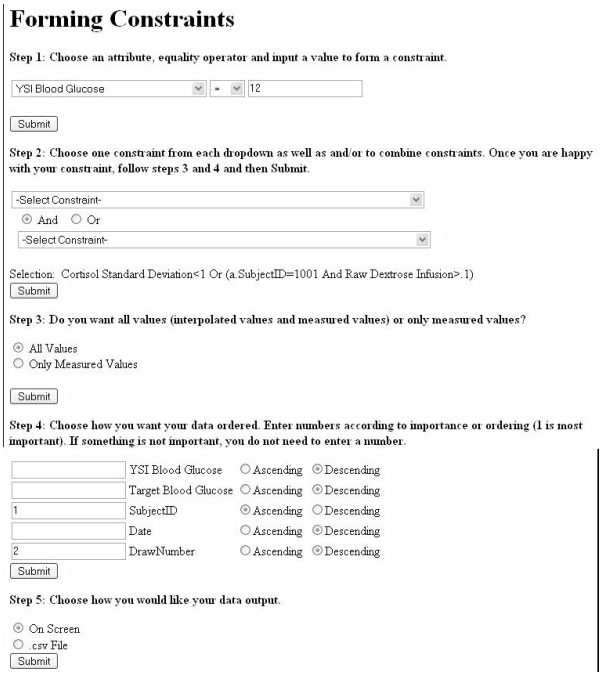

Fig. 4. Constraint Selection Page.

The first step is to build a constraint using two dropdown menus and a textbox. The first dropdown contains all of the variables in the study and the second has equality operators. After choosing the variable and equality operator, the users then input a number into the textbox and click on Submit. This populates both of the dropdowns under Step 2 with the newly formed constraint. The users can then continue to build single constraints in Step 1 to continue populate the dropdowns in Step 2.

Step 2 allows the users to compound constraints to build more robust queries by combining constraints using 'And' or 'Or'. The current constraint is shown below the dropdowns

so the users know which constraint they will be using. Once they are happy with the constraint, they will move onto Step 3. Step 3 allows the users choose whether or not they want all values or just the measured the values. Some of the values are interpolated, so by choosing only measured values, interpolated values will be excluded. Step allows the users to choose how they would like their data ordered. The users will enter numbers in the text boxes next to the variables they deem to be important for their data to be ordered by. The data will be ordered first by the number one and then following the numerical order after that. The final step is to choose whether the user wants the output to be On Screen or an Excel-compatible (.csv) file. Once they choose the appropriate radio button, they will click on the final Submit button on the page. This will generate the final query and query the database. The results will be output in the desired format and they can analyze the data as they see fit.

III. CONCLUSIONS

At the time of this writing, the upload and analysis functionalities are complete and functional. The upload tool is ready for users, but the analysis tool still needs some testing. Also, there are areas for future improvement on this tool.

A. Project Strengths

Overall, this project has been a success. We have completed both the upload and the analysis functionalities, the two main pieces of the project. There are many studies on the Medical Center's network, but none of the analysis tools comes close to being as robust as ours. The closest functionality allows a user to directly enter a query into a textbox to extract data from the database. Since the assistants generally have no SQL knowledge, this functionality is never used other than when the assistants directly ask the developer to run a query for them.

We worked under the assumption that the researchers had little to no database knowledge to make a user-friendly tool for uploading and extracting data. I believe the assistants will be able to use our tool to query the database without any outside help. This will save time for the researchers, as they will be able to ask the assistants for data and have it sent directly to them to analyze. Hopefully our tool will aid the researchers in this study in learning more about Type I Diabetes and its management.

B. *Areas for Improvement*

An important part of the Systems Engineering methodology that we did not complete due to time constraints is getting user feedback. We were introduced to one of the future users of the tool, a nurse working for the DTC, but we were never able to demonstrate our tool to her. I believe she would have been able to give us some very helpful feedback on where our tool is difficult to use, and some ways to make it more user-friendly for exactly what she will be doing.

At this point, only two of the tables in the database have automatic upload functionalities. The other tables were approved in the initial design but we were never asked to build a functionality to populate them. If the researchers really want to find a relationship between all four input variables using our tool, they will need to find a way to populate these tables.

Another enhancement that would be beneficial would be to have a way for the assistants to correct themselves if they make an error. They may load a file to the database and then later receive a revision to this file. As the system is currently, they cannot upload a new file for any subjects. The only way to do this would be to have a person connect directly to the database through MySQL and delete the applicable rows from the table. This enhancement would be nice to have, but is not a necessity.

ACKNOWLEDGMENT

We would like to thank Marc Breton, our technical advisor and contact and the DTC, for his help in the completion of this project. The team would also like to thank our Science, Technology and Society professors Bryan Pfaffenberger and Benjamin Cohen.

REFERENCES

[1] Cryer, P. E. (2002). Hypoglycaemia: The limiting factor in the glycaemic management of type I and type II diabetes. *Diabetologia, 45*(7), 937-948.

[2] Kovatchev, B. P. (2006). Counter-regulatory impairment and the effect of microvascular insulin transfer in type 1 diabetes mellitus.

[3] Hall, M. (1999). *Core servlets and JavaServer pages*.