Automatic Glycemic Alert System[1]

Soohee Lee, and Dongsoo S. Kim
Computer and Information Science, IUPUI (sl6@iupui.edu)
Electrical and Computer Engineering, IUPUI (dskim@iupui.edu)

Abstract— **A large randomized controlled study from Leuven, Belgium demonstrated that normalization of blood glucose levels using an intensive insulin infusion protocol (IIP) improved clinical outcomes in patients admitted to a surgical intensive care unit (ICU) (6). In the Leuven study, intensive insulin therapy (to maintain blood glucose target levels between 80 and 110 mg/dl) reduced ICU mortality by 42%. Based on this clinical evidence, there are increasing efforts worldwide to maintain strict glycemic control in critically ill patients. These efforts have developed standardized glucose control protocols, such as Insulin Infusion Protocols (IIP), that adjust insulin infusion rate tightly to meet the glucose target level. Such protocols are implemented for inpatients admitted to the hospital.**

Diabetes outpatients could maintain the same lower and stable glucose levels if they monitor their blood glucose level in regular base and apply similar glucose control protocol at home. However, it is not easy for patients to implement the protocol themselves without help of health care professionals. They cannot remember all the rules and get adjust insulin dosage whenever they check their blood glucose level. To help those under conventional insulin therapy, this article describes Automatic Glycemic Alert System that implements glucose control protocol in a central database as rules and instructions. The rules in the database indicate how much insulin (or carbohydrate in case of hypoglycemia event) a patient needs to take for each range of glucose level. The rules are specific for each individual patient. When the system gets a patient's blood glucose test result from wireless glucose tester, a system trigger is automatically activated to compare the result to the rules in the database to see if the result falls into one of the rules. If it does, the trigger generates an instruction that tells the patient how much insulin or carbohydrate she should take to control glucose level as the target range. Then, it delivers the instruction to the patient's mobile phone as a text message or regular Email. This experiment shows the possibility to use modern technology as a communication bridge between healthcare professional and outpatients. With the help of technology, healthcare professional can outreach to their outpatients and take care of them as they were in the hospital.

I. INTRODUCTION

In 2001, a large randomized controlled study from Leuven, Belgium, demonstrated that normalization of blood glucose levels using an intensive insulin infusion protocol (IIP) improved clinical outcomes in patients admitted to a surgical intensive care unit (ICU) (6). In the Leuven study, intensive insulin therapy (to maintain blood glucose target levels between 80 and 110 mg/dl) reduced ICU mortality by 42% and also reduced the incidence of bloodstream infections, the incidence of acute renal failure, the need forprolonged ventilatory support, and the length of stay in ICU. Strict glycemic control appears to be beneficial in other intensive care settings as well.

Based on this clinical evidence, there are increasing efforts worldwide to maintain strict glycemic control in critically ill patients. These efforts have developed standardized glucose control protocols, such as Insulin Infusion Protocols (IIP), that are practical and easily implemented by nurses without the need of frequent physician input.[5] In the protocols, nurses are given standardized instructions to follow when they treat diabetes patients to achieve the target glucose level. Yale Insulin Infusion Protocol [4] below is one of the samples of Insulin Infusion Protocol (IIP). It gives nurses instruction how to change insulin infusion rate based on blood glucose monitoring result.

A. YALE INSULIN DRIP PROTOCOL

The following insulin drip protocol is intended for use in hyperglycemic adult patients in an ICU setting, but is not

[1] This work was supported in part by the National Science Foundation under grant CCF-0514975.

specifically tailored for those individuals with diabetic emergencies, such as diabetic ketoacidosis (DKA) or hyperglycemic hyperosmolar states (HHS). When these diagnoses are being considered, or if BG = 500 mg/dL, an MD should be consulted for specific orders. Also, please notify an MD if the response to the insulin drip is unusual/unexpected, or if any situation arises that is not adequately addressed by these guidelines.

B. Initiating an Insulin Drip

1) INSULIN INFUSION: Mix 1 U Regular Human Insulin per 1 cc 0.9 % NaCl. Administer via infusion pump (in increments of 0.5 U/hr).
2) PRIMING: Flush 50 cc of Insulin/NS drip through all IV tubing, before infusion begins (to saturate the insulin binding sites in the tubing)
3) TARGET BLOOD GLUCOSE (BG) LEVELS: 100-139 mg/dL
4) BOLUS & INITIAL INSULIN DRIP RATE: Divide initial BG level (mg/dL) by 100, then round to nearest 0.5 U for bolus AND initial drip rate. Examples:
    a. Initial BG = 325 mg/dL: 325 ÷ 100 = 3.25, rounded ? to 3.5: IV bolus 3.5 U + start drip @ 3.5 U/hr.
    b. Initial BG = 174 mg/dL: 174 ÷ 100 = 1.74, rounded ? to 1.5: IV bolus 1.5 U + start drip @ 1.5 U/hr.

C. Fingerstick (FS) Blood Glucose Monitoring

1) Check FS hourly until stable (= 3 consecutive values in target range)
2) Then check FS q 2 hours; once stable x 12-24 hours, FS checks can be spaced to q 4 hours if:
    a. No significant change in clinical condition AND b) No significant change in nutritional intake
3) If ANY of the following occur, consider the temporary resumption of hourly FS monitoring, until BG is again stable (= 2-3 consecutive BG values in target range).
    a. Any change in insulin drip rate (i.e. BG out of target range)
    b. Significant changes in clinical condition
    c. Initiation or cessation of pressor therapy
    d. Initiation or cessation of renal replacement therapy (hemodialysis, CVVH, etc.)
    e. Initiation, cessation, or rate change of nutritional support (TPN, PPN, tube feedings, etc.)

D. Changing the Insulin Drip Rate

1) If BG < 50 mg/dL

D/C INSULIN DRIP: Give 1 Amp (25 g) D50 IV; recheck BG q 15 minutes. When BG = 100 mg/dL, wait 1 hour, then restart insulin drip at 50% of original rate.

2) If BG 50-74 mg/dL

D/C INSULIN DRIP: If symptomatic (or unable to assess), give 1 Amp (25 g) D50 IV; recheck BG q 15 minutes. If asymptomatic, give 1/2 Amp (12.5 g) D50 IV or 8 ounces Juice; recheck BG q 15-30 minutes. When BG = 100 mg/dL, wait 1 hour , then restart drip at 75% of original rate

3) If BG = 75 mg/dL

STEP 1: Determine the CURRENT BG LEVEL - identifies a COLUMN in the table:

BG 75-99 mg/dL	BG 100-139 mg/dL	BG 140-199 mg/dL	BG = 200 mg/dL

STEP 2: Determine the <u>RATE OF CHANGE</u> from the prior BG level - identifies a <u>CELL</u> in the table. Then move right for **<u>INSTRUCTIONS</u>.** Note: If the last BG was measured 2 – 4 hrs before the current BG, calculate the hourly rate of change. <u>Example</u>:

If the BG at 2PM was 150 mg/dL and the BG at 4PM is now 120 mg/dL, the total change over 2 hours is -30 mg/dL; however, the <u>hourly</u> change is –30 mg/dL ÷ 2 hours = -15 mg/dL/hr.]

BG 75-99 mg/dL	BG 100-139 mg/dL	BG 140-199 mg/dL	BG = 200 mg/dL	Instructions *
		BG ? by > 50 mg/dL/hr	BG ?	? **DRIP by "2?"**
	BG ? by > 25 mg/dL/hr	BG ? by 1-50 mg/dL/hr *OR*	BG UNCHANGED OR	? **DRIP by "?"**

		BG UNCHANGED	BG ? by 1-25 mg/dL/hr	
BG ?	BG ? by 1-25 mg/dL/hr, BG UNCHANGED, *OR* BG ? by 1-25 mg/dL/hr	BG ? by 1-50 mg/dL/hr	BG ? by 26-75 mg/dL/hr	***NO DRIP CHANGE***
BG UNCHANGED OR BG ? by 1-25 mg/dL/hr	BG ? by 26-50 mg/dL/hr	BG ? by 51-75 mg/dL/hr	BG ? by 76-100 mg/dL/hr	? **DRIP by "?"**
BG ? by > 25 mg/dL/hr *see below***	BG ? by > 50 mg/dL/hr	BG ? by > 75 mg/dL/hr	BG ? by > 100 mg/dL/hr	**HOLD DRIP x 30 min, then ? DRIP by "2?"**

**D/C INSULIN DRIP; vBG q 30 min; when BG = 100 mg/dl, restart drip @75% of original rate.

*** CHANGES IN DRIP RATE ("?") are determined by the current drip rate:**

Current Drip Rate (U/hr)	? = RateChange (U/hr)	2? = 2X RateChange (U/hr)
< 3.0	0.5	1
3.0 - 6.0	1	2
6.5 - 9.5	1.5	3
10 - 14.5	2	4
15 - 19.5	3	6
20 - 24.5	4	8
= 25	= 5	10 *(Consult MD)*

A Diabetes Study in Germany shows self monitoring of blood glucose clearly reduces the occurrence of secondary diabetes complications (7). Diabetes patients could maintain the same lower and stable glucose levels if they monitor their blood glucose level in regular base and apply similar glucose control protocol at home. However, it is not easy for patients to implement the protocol themselves without help of health care professionals. Even if physicians may give them instruction how to control hypo/hyperglycemia, it is a challenge for patients to remember all the rules and get adjust insulin dosage whenever they check their blood glucose level. Serious diabetes patients may get help from insulin pump that continuously monitors glucose and adjust background insulin dosage based on the result. However, they often find insulin pumps are expensive, uncomfortable to wear, sometimes cause injection sites infected, and have mechanical problems such as leakage. Due to the issues, many diabetes patientsare hesitant to use it unless their blood glucose level is really hard to control. Majority of diabetes patients are still dependent on the conventional insulin therapy and they need a tool that helps them monitor blood glucose, implement glucose control protocol easily at home, and guide them to adjust insulin dosage based on their glucose level.

To help those under conventional insulin therapy, this article describes Automatic Glycemic Alert System that implements glucose control protocol in a central database as rules and instructions. The rules in the database indicate how much insulin (or carbohydrate in case of hypoglycemia event) a patient needs to take for each range of glucose level. The rules are specific for each individual patient. When the system gets a patient' s blood glucose test result from wireless glucose tester, a system trigger is automatically activated to compare the result to the rules in the database to see if the result falls into one of the rules. If it does, the trigger generates an instruction that tells the patient how much insulin or carbohydrate she should take to control glucose level as the target range. Then, it saves the instruction in a database table. A java program run as a background process and checks the database table every one minute to deliver the instruction to the patient' s mobile phone as a text message or regular Email.

Using the Internet, wireless network, database, and SMS (Short Message Service) technology, diabetes patients receive the exact same instruction that physician would give when their blood glucose level is out of the target range.

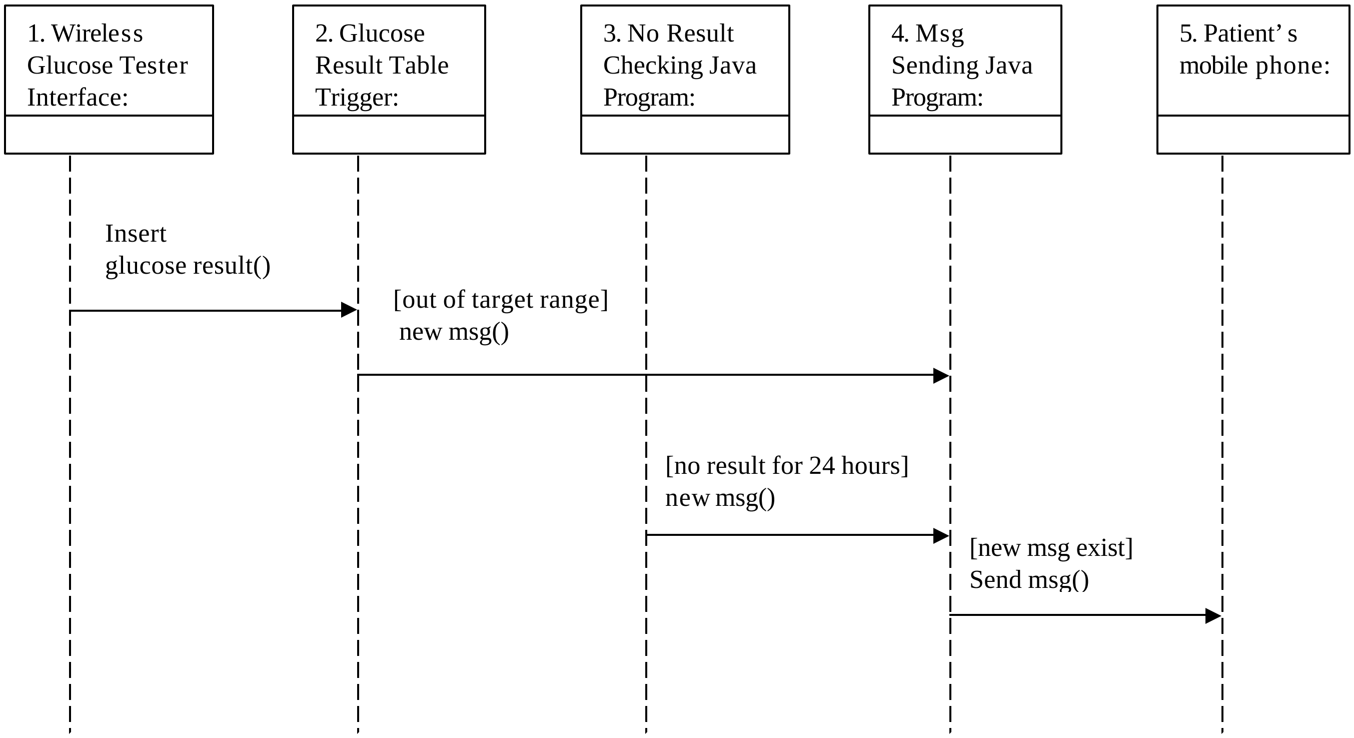


Figure 1. Sequence diagram of the proposed system

The instruction is the glucose control protocol implemented as a built-in database component. It guides patients to adjust insulin dosage based on their glucose level. The system also checks anypatients who have no glucose test result during a specific period of time and sends an alert text message that urges the patient to monitor glucose test. This functionality helps diabetes patients to monitor blood glucose level in regular base.

II. DEVELOPMENT ENVIRONMENT

A ECE department Sun Solaris server is used as a database server and application server. MySQL database is used as database. Development languages are Java and PL/SQL for trigger.

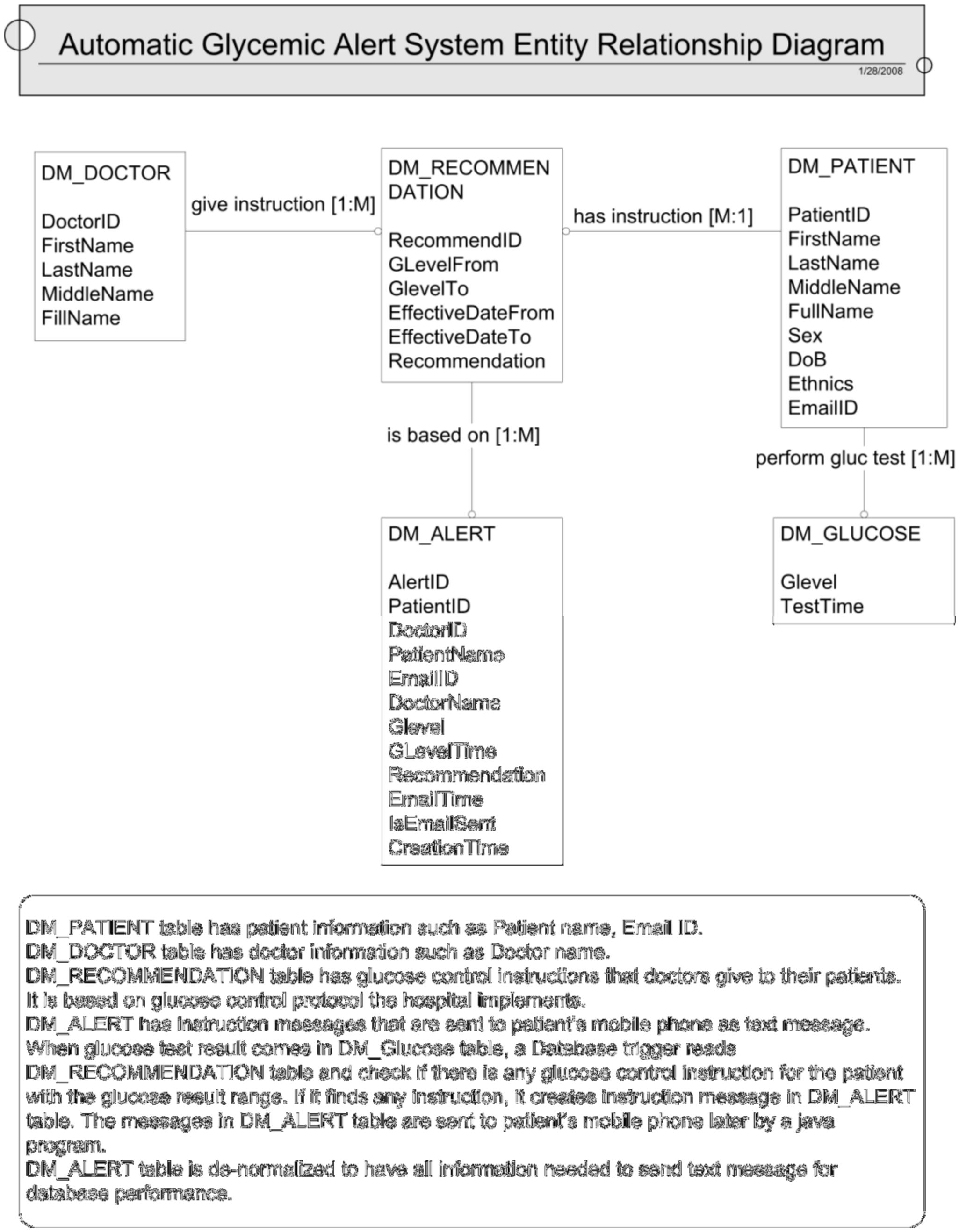
Category	***Product Name & Version***	***Description***
Hardware / OS	Sun Solaris	ECE department Unix machine (ece.sl.iupui.edu) is used as database server and application server
Database	MySQL 5.0.24	MySQL, freeware DBMS (Data Base Management System), is downloaded from the Internet & installed. MySQL version 5.0 is installed to use PL/SQL.
Programming Language	JAVA2 SE 5.0, PL/SQL	Java Standard Edition Development Kit is downloaded from the Internet & installed. MySQL PL/SQL is used to develop database trigger

III. UML (Unified Modeling Language)

The Sequence Diagram below describes how a patient's glucose result activates database trigger to create glucose control instruction and how the instruction is delivered to the patient as a text message.

1) Wireless Glucose Tester sends glucose test result of a patient to the server. The interface program saves the result to the database.
2) When the test result is inserted in the database table, a database trigger is activated automatically. The trigger compares the test result to glucose control protocol rules stored in the database. The rules are prescribed by p hysicians and they are specific to each patient. If there is any rule for the test result range (e.g. between 200 to 220), the trigger creates an instruction and save it in a table.
3) A background java program checks if there is any patient who has no glucose test result for last 24 hours. If there is, it creates a reminding message and save it in a table.
4) A java program runs every minute and checks if there is any new messages created from step2 or step 3. If there is, it sends the message to patient's mobile phone as a text message
5) Patients get glucose control instruction by mobile phone and can adjust their insulin dosage based on the instruction. Or they get a message that urge them to perform glucose test.

IV. Entity Relationship Diagram



V. Directory Structure

The directory described below is under $CATALINA_HOME which is the tomcat home directory.

directory	***sub dir 1***	***sub dir 2***	***sub dir 3***	***contents***
diabetes				Main directory for the project.
	DataGen			Test data generation programs.
	MySQL			MySQL scripts
	WEB-INF			
		lib		jar files.

		classes		Java source files developed.
			diabetes	java class files (bytecodes)

VI. PROGRAM & SCRIPT DESCRIPTION

1. Database scripts. (directory path: $CATALINA_HOME/diabetes/MySQL)

The scripts below is used to create tables and generate test data.

-. dm_Trigger.sql

Create "after insert" trigger on dm_glucose table. The trigger is automatically activated after a row is inserted in dm_glucose table. It compares the glucose level in dm_glucose table against the recommendations of the patient in dm_recommendation table. If there is any recommendation for the glucose level, it inserts the recommendation into dm_alert table, so it can be emailed to the patinet' s mobile phone later.

-. genTriggerTest.sql

To test dm_glucose trigger.

-. genNoTest.sql

To get patients who has no data in dm_glucose table during specific priod.

-. regenDB

Regenerate diabetes database. This script runs the following scripts: dropTable.sql, createTable.sql, dataLoad, and updatePswd.sql. These scripts can be run separately if necessary.

-. createTable.sql

Create tables used in this application.

-. dropTable.sql

Drops tables and integrity constraints (e.g. primary key or foreign key constraint).

-. dataLoad

Upload data to the tables. It can be used for test data or history data. If the table layout is changed, this script should be changed too.

2. .java files. (directory path: $CATALINA_HOME/diabetes/WEB-INF/classes)

These are the java source files developed as bean programs. The class files (bytecode file), which are generated by compiling these source files, are located under $CATALINA_HOME/diabetes/WEB-INF/classes/diabetes because all the programs are defined as a member of ' diabetes' package (This means that each program includes ' package diabetes;' line in the beginning of the program). If the package name is changed, the directory structure can be changed based on the package name. The compile command is ' javac –d . sourceFileName' .

-. AlertGlResult.java

This is a java class source file that reads dm_alert table and send email to patient' s mobile phone through SMTPS. It updates 'email sent flag' and 'time' columns of dm_alert table after email is sent. Timer object is used to run the class every minute by default. Change time duration if needed.

-. AlertNoTest.java

This is a java class that checks any active patients who have no glucose test result done within 24 hours and sends email to them. Timer object is used to run the class every minute by default for testing. Change time duration as appropriate value.

-. DBConnection.java

This is a db connection class. All programs which need to access ‘ diabetes’ database use the class.

VII. TEST & RESULT

1. Prerequisite: This is prerequisite environmental setting to test database trigger and alert functionality.

-. MySQL JDBC Driver class

JDBC driver class is needed to access MySQL database. JDBC for MySQL is distributed as a .jar file. For example, mysql-connector-java-5.0.7-bin.jar is JDBC driver for MySQL version 5.0.7. The jar file should be located on $CATALINA_HOME/diabetes/WEB-INF/lib/.

-. Java Classes (.java file)

Java classes need to be located in $CATALINA_HOME/diabetes/WEB-INF/classes/. The source files need be compiled with the commend: javac –d . ‘ soruceFileName’ The byte code will be generated under the package directory, which is ‘diabetes’ in this project.

2. Testing: It shows how to test Trigger and get Glucose control instruction as a text message

- Execute $CATALINA_HOME/diabetes/MySQL/mlogin to login to MySQL
  $CATALINA_HOME/diabetes/MySQL> mlogin

- Test patient ‘2’ has recommendation in dm_Recommendation table.

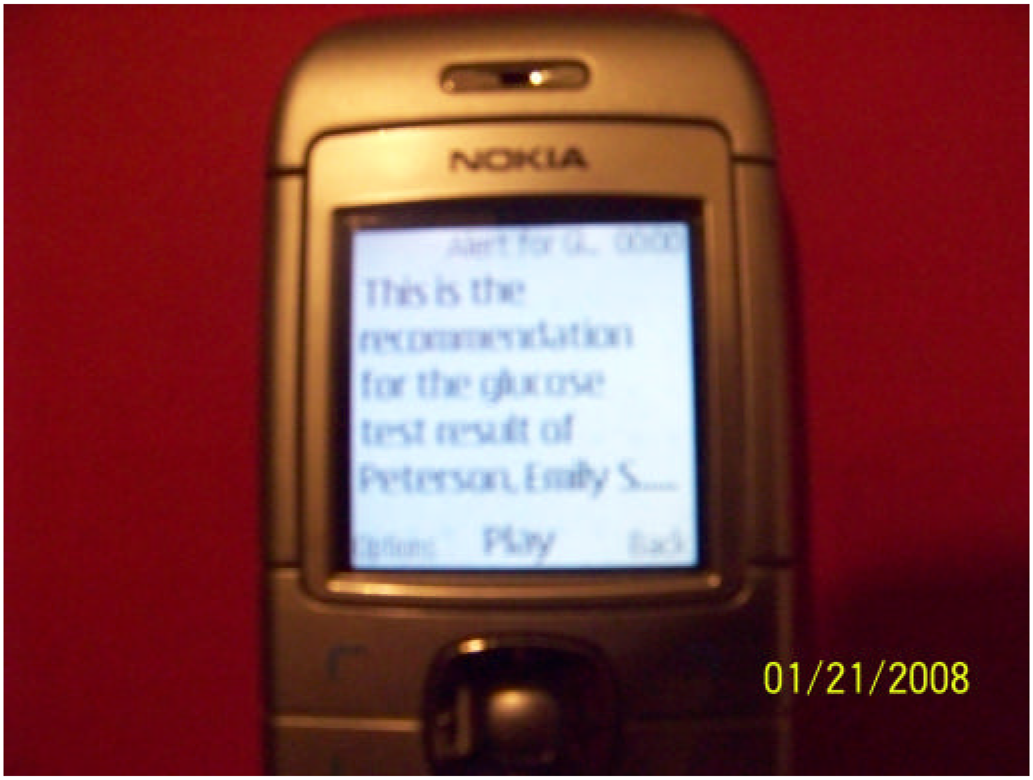
```
mysql> select gLevelFrom, gLevelTo, recommendation
    -> from dm_recommendation
    -> where patientID=2;
+------------+----------+---------------------+
| gLevelFrom | gLevelTo | recommendation      |
+------------+----------+---------------------+
|        140 |      159 | Orange Juice 1 cup  |
|        160 |      179 | Inject Glucose 5mg  |
|        180 |      199 | Inject Glucose 10mg |
|        200 |      999 | Inject Glucose 20mg |
+------------+----------+---------------------+
4 rows in set (0.00 sec)
```

- Update her email ID with desired email ID.
  mysql> update dm_patient set emailID='8125813333@tmomail.net' where patientID=2;

- Insert glucose test result into her result table. The glucose result should fall into one of glucose range in dm_recommend table.
  mysql> insert into dm_glucose (glucoseID,patientID,testerID,testTime,gLevel)
  values (null, 2, 2, NOW(),200);

Instead of typing Insert SQL above, SQL script can be executed to insert the same glucose result as below.
mysql> \. getTriggerTest.sql

3. Result: Database trigger on dm_glucose checked dm_recommendation table and created instruction in dm_alert table automatically when glucose result is inserted into dm_glucose table. Java program sent the instruction to destination SMTP server immediately. In most of cases, it took 10-15 minutes to actually get the instruction in mobile phone as a text message. The delivery time seems to vary depending on where the destination mobile phone is located. Picture below is the instruction received in mobile phone.



VIII. Conclusion and Future work

This project experiments how we can implementglucose control protocol for diabetes outpatients as they were in the hospital. We can see the possibility by using Internet, Wireless network, SMTP, and SMS (Short Message Service) technology. Patient's blood glucose monitoring data is gathered in the database using Internet and wireless network. Glucose control protocol is implemented asa built-in database component. When glucose result comes in the database, a automatic database trigger checks the glucose level against the built-in protocol to see if any action needs to be done to control patient's glucose level as target range. If it does, a java program sends SMS message to the patient's mobile phone to remind the patient specific instruction to follow. In this way, diabetes patients can receive the exact same instruction that doctors would give to their inpatients when their blood glucose level is out of target range. The instructions guide patients to adjust insulin dosage based on their glucose level. This experiment shows the possibility to use modern technology as a communication bridge between Healthcare professional and outpatients. With the help of technology, healthcare professional can outreach to their outpatients and take care of them as they were in the hospital.

The lists below are works that may need to be done in the future as this project continues.

1. It was easy to find Insulin Infusion Protocol (IIP) for inpatients. However, standardized glucose control protocol for outpatients is hard to find. Most of diabetes outpatients are dependent on the recommendation from their own doctor. Standardized glucose control protocol needs to be developed for outpatients in the future.

2. Since Insulin management protocol needs to be customized for individual patient, a user interface that allows doctors to give specific instruction for each patient is needed. The system may need to implement electronic signature technology to verify the doctors and their confirmation.

3. Right now glucose control protocol is implemented as hard-coded rules in the database. Eventually, it may need to be evolved as knowledgebase database. Depending on the change of patient' s glucose level, the rules need to be changed automatically. Instead of updating the hard-coded rules, doctors may set "meta-rules", the rules that manipulate hard-coded rules depending on the glucose test result.

4. Several administrative systems are needed to implement the system in practice. For example, administrators need user interface programs that allow them to register glucose tester, Insulin management rules etc.

IX. REFERENCES

[1] Diabetes Mellitus. At http://www.aace.com/meetings/ams/2007/pdfs/2007AM-Abstracts.pdf

[2] Implementation of a Safe and Effective Insulin Infusion Protocol in a Medical Intensive Care Unit. At http://care.diabetesjournals.org/cgi/content/full/27/2/461

[3] Diabetes 2007. At http://cme.yale.edu/diabetes/2007/ADA-Newsletter-1-June23.pdf

[4] Yale Insulin Infusion Protocol at http://care.diabetesjournals.org/content/vol27/issue2/images/large/zdc0020409550001.jpeg

[5] Intensive Glucose Management in Critically Ill Patients Improves Patient Outcomes. At http://www.cop.ufl.edu/wppd/research/sample06a.pdf

[6] Intensive Insulin Therapy in Critically Ill Patients. At http://content.nejm.org/cgi/content/abstract/345/19/1359?ijkey=73d8e7be10400bd6afbc7a81716a89435a952130&keytype2=tf_ipsecsha

[7] Meal-Related Structured Self-Monitoring of Blood Glucose. At http://care.diabetesjournals.org/cgi/content/full/25/11/1928

X. APPENDIX

TABLE DESCRIPTION

a. dm_recommendation

contains doctor' s recommendation for each glucose level

Field	Type	Null	Key	Default	Comments
recommendID	Auto_increment	NO	PRI	NULL	auto_increment
patientID	int(10) unsigned	NO	FK		FK to dm_patient
doctorID	int(10) unsigned	NO	FK		FK to dm_doctor
gLevelFrom	smallint(5) unsigned	NO			Glucose range low value
gLevelTo	smallint(5) unsigned	NO			Glucose range high value
effectiveDateFrom	datetime	NO			Effective from date
effectiveDateTo	datetime	No		NULL	Effective to date
recommendation	Varchar(200)	YES		NULL	recommendation
Attribute1 - 10	varchar(100)	YES		NULL	

b. dm_alert

It contains doctor' s recommendation sent to patient' s cell phone as text message

Field	Type	Null	Key	Default	Comment
alertID	int(10) unsigned	NO	PRI	NULL	auto_increment
patientID	int(10) unsigned	NO	MUL		FK to dm_patient
recommendID	int(10) unsigned	NO			FK to dm_recommendation
doctorID	int(10) unsigned	NO	MUL		FK to dm_doctor
patientName	Varchar(100)	NO			Patient name
emailID	datetime	NO		NULL	Patient Email id
doctorName	Varchar(100)	NO			Doctor name
gLevel	smallint(5) unsigned	NO		NULL	Patient' s glucose result
gLevelTime	datetime	NO			Test time
recommendation	Varchar(200)	NO			recommendation
emailTime	datetime	YES			The time email sent
isEmailSent	Char(1)	YES			Whether email sent or not
creationTime	datetime	NO			The time record created
attribute1 - 10	varchar(100)	YES		NULL	

c. dm_doctor

contains doctors' information.

Field	Type	Null	Key	Default	Comments
doctorID	int(10) unsigned	NO	PRI	NULL	auto_increment
firstName	varchar(40)	NO	MUL		
lastName	varchar(40)	NO	MUL		
middleName	varchar(40)	YES		NULL	
fullName	varchar(100)	NO			
logonID	varchar(20)	YES		NULL	
password	varchar(20)	YES		NULL	Should be encrypted using 'PASSWORD' procedure
attribute1 - 10	varchar(100)	YES		NULL	

d. dm_glucose

contains the glucose level test information for a specific patient.

Field	Type	Null	Key	Default	Comment
glucoseID	int(10) unsigned	NO	PRI	NULL	auto_increment
patientID	int(10) unsigned	NO	MUL		FK to dm_patient
testerID	int(10) unsigned	NO			FK to dm_tester
testTime	datetime	NO			Glucose test date and time
gLevel	smallint(5) unsigned	NO			Glucose level checked at testTime
attribute1 - 10	varchar(100)	YES		NULL	

e. dm_patient

contains patient information.

Field	Type	Null	Key	Default	Comment
patientID	int(10) unsigned	NO	PRI	NULL	auto_increment
firstName	varchar(40)	NO	MUL		
lastName	varchar(40)	NO	MUL		
middleName	varchar(40)	YES		NULL	
fullName	varchar(100)	NO			
sex	char(1)	NO			
DOB	date	NO			
ethnics	varchar(20)	NO			

doctorID	int(10) unsigned	NO	MUL		FK to dm_doctor
criterialD	smallint(5) unsigned	NO			FK to dm_criteria
cellPhone	varchar(15)	YES		NULL	
emailID	varchar(40)	YES		NULL	Include email id or cellphone number (e.g. 8123334444@tmobile.com) to notify alert to patient or guardian.
attribute1 - 10	varchar(100)	YES		NULL	