The Simulation – Based Solution to Detection by Collectively Decomposing Factors of IMU and Image Process Data in the Indoor Environment

J.A.D.C.Anuradha Jayakody [1],R.G.P Dulshan[2], D.C Meegahawatta[3] , H.M.K.S.B Herath[4], L.P.A.D Pathirathne[5]

Department of Information Technology, Sri Lanka Institute of Information Technology (SLIIT), Malabe, Sri Lanka.

[1]anuradha.j@sliit.lk, [2]dulshanqw@gmail.com, [3]chamikaradinuka@gmail.com, [4]kalanaherath15@gmail.com, [5]ashan.dileepa@gmail.com

ABSTRACT

Existing indoor navigation system face with many different technical and usability problems because of the localization. In this paper presents indoor navigation simulation-based solution that can apply to the real implementation. The proposed system has several benefits and has the potential to increase the usability of the scheme.

Navigation is the process of monitoring and controlling the movement of and item from an origin to a destination along the path. Navigation system provides reading monitoring and updating the movement of one's position and guiding by intelligible visuals. Audible or Tangible means while she is traveling on an intended route.

In this simulation movement of the user is shown by the navigation and based on the simulation human movement can identify the path, location, the remaining distance to their final destination. Moreover, this will be critical for the user to determine their exact path. Any number of user can use this at the same time, and it has the facility to simulate the crowd sourced.

Environment. Further, this research work focuses database optimization on reducing access time as an optimized solution.

Based on the simulated results authors plan to develop an algorithm that facilitates to indoor navigation with localized information and plan to discuss test results with evaluation.

Keywords- **IMU, REST, ath_H_ID, ImgD , Labelname(LN),PlaceDescription(PD),StartingX (SX),EndingX(EX),StartingY(SY),EndingY(EY) ,FloorID(FID),BuildingID(BID).**

I. INTRODUCTION

This context is providing a complete idea about Indoor navigation and database optimization mechanisms. This research can be described three main components. Those are,

- Data Collection
- Data storing
- Utilizing stored data

As an initial step for the research, it needs a huge data collection first. That data collection will be the input for the database optimizing part. Therefore the research team has populated this huge data collection by using two main techniques which commonly available in current indoor navigation systems.

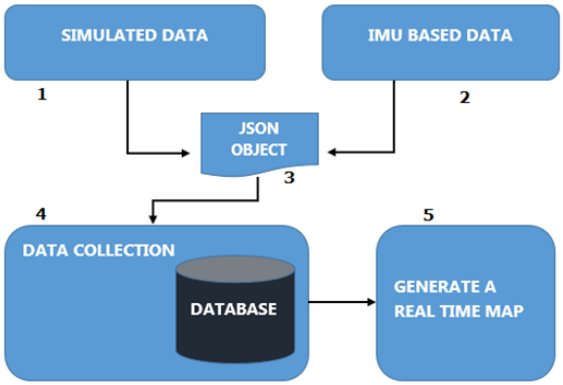

Figure 1: Overall System Architecture

1. Simulated Data

Under this component, it describes a simulated indoor navigation model which generates hundreds of records within few seconds. It will produce live data collection base on the user's requirement. What the user has to do is only select the number of user and their destinations. After that, the simulator navigates selected user to their destination, and meanwhile, it collects all the information which belong to user's each and every movement.

Ex: User current x coordinates, User current x coordinates, R-value, Theta Value, Obstacle detection, environment details .(door info, wall info)

2. IMU-Based data

Under this component, it describes real-time data. Here a real-time navigation based data collection will be generated on behalf of simulated data collection. It uses IMU (inertial measurement unit) techniques to generate those real-time data collection. What the user has to do is only walking along the path with his smart phone. Smart phone itself identify the user's current coordinates and step counts. Base on IMU data it produces same data collection as simulated data collection.
Ex: User current x coordinates, User current x coordinates, R-value, Theta Value, Obstacle detection, environment details. (door info, wall info using image processing)

3. JSON Object

In this research, it generates huge data collection by two ways. That data should be accessed the Database at the same time without any delay. Here a JSON object is used in order to make the process speed up and real time.

Why JSON?

- JSON is a much simpler.
- Smaller message size
- More structural information in the document
- The speed of processing.
- JSON is easier to read.
- JSON requires fewer tags

In every database call, all the values will be converted to a JSON array and send them to the database through the web service.

4. Data Collection

In this component, it will optimize and store all the data comes from the user. (May be the simulator or a real user). For the database optimization part, it has been used several techniques.
Ex:

- Indexing
- Memcaching
- Profiling
- Costing
- Stored procedures
- Generate a real time map

Up to this level, we have collected a huge data set, and that data set has been optimized by using several optimizing mechanisms. Therefor at this step, it is going to be utilized that optimized data collection by generating a real time map. Real time in the sense, it can automatically update the environmental changes time to time.
Ex: Door information, wall information, and moveable object information, etc.

II COLLECT DATA FROM SIMULATOR

In the simulator, user can select the number of users and their starting point and destinations respectively. Then once the app user started the navigation process, it will collect each and every movement information of every user. Following variables will be captured during the navigation process.

1. User's current x coordinates.
2. User's current y coordinates.
3. R-value
4. Theta value
5. Is obstacles ahead?
6. Obstacle details.
7. Environment details.

1. User's current X and Y coordinate

The map of the simulator will hold the following ratio with the real world.

Map ratio : 8 pixel = 0.8 m

Therefor every time a marker passes 8 pixels, it will be counted as 1 step. Step count will be captured according to that scenario.

2. User is angle. (Theta value)

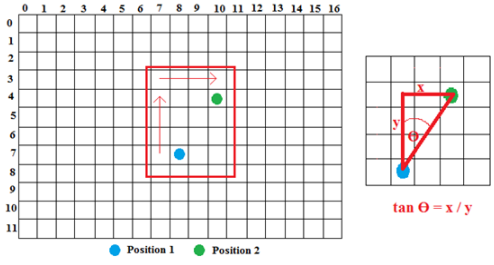

Figure 2 : User's Angel

Let's say the user moved position 1 to position 2 as mentioned in the above picture. Then the Theta value will be calculated as follows.

Calculation:

$$\text{Tan } \Theta = (10 - 7) / (8 - 4)$$
$$= \tan^{-1} [0.75]$$
$$= 36.869^{0}$$
$$= 37^{0}$$

3. *Obstacle detection*

Normally once a user starts his or her navigation it keeps an interested area. It means, the user well aware about all the movable and non-movable objects with in that area as mentioned in the following image.

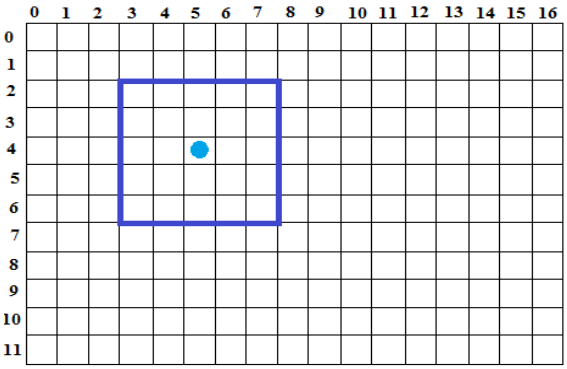

Figure 3 : Obstacle Detection

Once the user detects some object within his or her interested area, it identifies whether it is a movable object or non-movable object.

If it is a movable object, the user considers it as obstacles and calculates the distance and angle of the obstacles based on user's current position. At the same time, that information will be sent to the database as well.

If it is a non-movable object user identifies it is a door object. There can be multiple objects within the user's interested area. Still, it identifies each object and updates their status. (Status = whether the door is open or not)

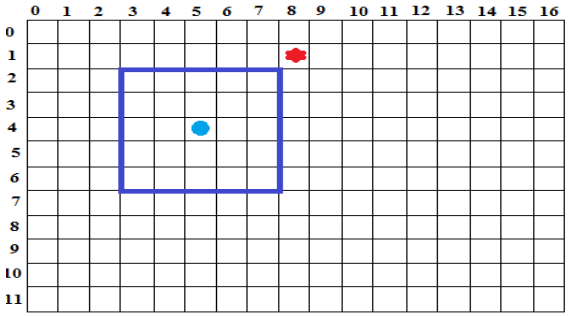

Figure 4 : Obstacle Identification Non Moveable

4. *Data communication between simulator and the server*

The communication between the server and the Navigation system is done using the JSON (JavaScript Object Notation). It is a lightweight data interchange format. It is based on a subset of the JavaScript Programming Language. To construct

HTTP requests and responses plain text or JavaScript Object Notation (JSON) is used. However, to send multicast messages must use JSON. Multicast messages mean the ability to send the same message to multiple devices simultaneously.

III. MAP GENERATION

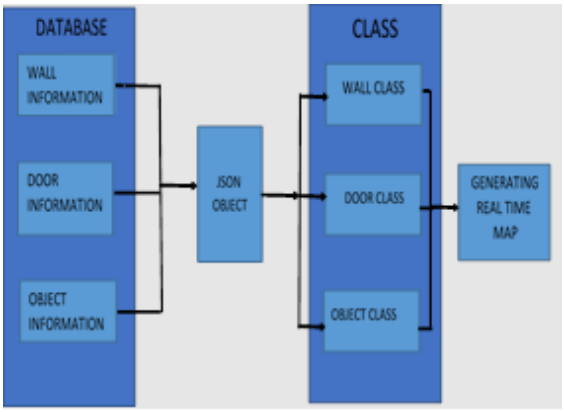

Figure 5 : Map generation overview

In the above diagram is explained the how the data is taken from the database. In the database, there is three tables called wall info, door info, object info, and we collect that data through the JSON object and build the three class call wall class, door class, object class. So final using that data and generate the map. All the object is shown on the map according to the given data.

This section introduces the methods and techniques that we have planned to follow in order to achieve our goal in this research project.

While navigation systems for outdoor environments are already available, navigation within buildings still poses a challenge. It's difficult to create a map in accordance with navigation. This map is created using the wall information of x, y coordinates. Below mention is some steps in accordance.

- Gathering data from the database.
- Identify the wall information related to x and y coordinates.
- This identifies the situation related to physical.(doors with x, y coordinates as follows if a door is closed it represented by red if a door is closed it represented by green)
- When the users move, the object in the building is automatically identified and plotted on the map.

When the map is created, these four fact is mainly taken into consideration and time to time the database access to create a map. The outcome is the real time map. The system overview is mainly based on client-server architecture. The map and the navigation pointer will be displayed on the client side with the smart device. All the sensor data will be collected in the actual environment by the smart device and generate the routing data from the virtual environment in a desktop application. The desktop application and mobile device are exchanging data through a communication protocol. The sensor information will be partially processed by the application and send them to the server to build the map.

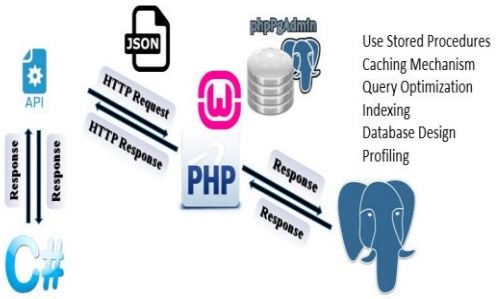

Figure 6 : Overview of the database optimization

IV. DATABASE OPTIMIZATION

Figure 6 diagram is explained that the overall view of how the database is optimized with the help of the API created by using c# dll, web service written by using PHP and PostgreSQL database management system. Several Technics such as stored procedures, caching mechanism, changing the database design, indexing, query rewriting, functions and profiling are used to optimize data inside database management System.

According to Figure 6 diagram, C# application (simulator) and PostgreSQL database are there. Web service based REST architecture is used to transfer data between the C# application and the PostgreSQL database. REST is web standards based architecture and uses HTTP Protocol for data communication. It revolves around resource where every component is a resource, and a resource is accessed by a common interface using HTTP standard methods. REST is almost always going to be faster and leading to lower bandwidth.

API is created using C# dll to interact C# application and web service. This dll consists of the database connector and the API functions (insert, update, and delete data within the table) to send and received required information. That data should be accessed the Database at the same time without any delay. For that purpose a JSON object is used in order to make the process speed up and real time. Main queries are executed inside postgrSQL database management system by using stored procedures so that it can be shared by a number of programs.

Reading data from a disk is still much slower than reading data from memory. In order to maintain an overall speed of the real-time database, memory is used to cache the most recent read data so next time same data is requested, the system will read directly from memory instead of disk.

The architectural overview (Figure.7) shows how the database schema organizes tables to store indoor data. In the initial stage, it includes features such as database connections, function call, and function return. Those function should occur respectively to fulfill the manipulation of the relation.

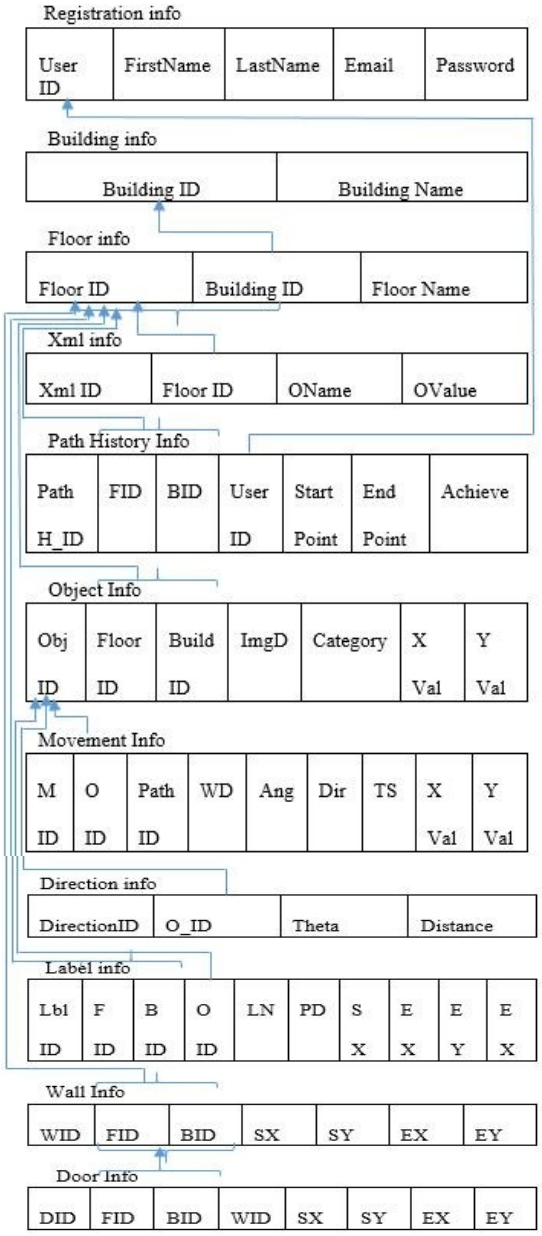

Figure 7: Design to present an organization of indoor navigation data

The proposed schema (Figure.7) clearly shows the table relationships and the primary and foreign keys to denote the information of the relational tables that can be used to develop an indoor map that can be support the people who are keenly waiting for using it in the indoor premises. The "object info" table has "Object id" as a master key which is generated automatically by the schema. It follows that object description field. An object id should refer to the "Label info" table attribute "Object id" as a foreign key and then get the received label information. The "Movement info" table should need to access the "Path history info" table to get the routing details. The time stamp attribute should refer to the "Movement info" table and "Path history id" as a foreign key. An "Object id" of the "Object Info" table should refer to the "Direction info" table to find the direction details through "Object id" of "Direction Info" table. "Floor info" has "floor id" and "building id" as a composite master key. That master key goes as a foreign key to "building info". Every floor has a specific xml file to map details. "Wall info" has "wall id," "floor id" and "building id" as a composite master key and that master key goes as a foreign key to "Door Info" table.

V. CONCLUSIONS & FUTURE

There is a great number of scopes for future improvements to our current implementation that can enhance both the performance and the user experience of the application.

ACKNOWLEDGEMENT

This work has been supported by Sri Lanka Institute of Information Technology, Malabe, Sri Lanka.

REFERENCES

[1] Y. Alon, A. Ferencz, and A. Shashua. Off-road path following using region classification and geometric projection constraints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2006*, volume 1, pages 689–696, June 2006.

[2] Levente Bagi. Pedometer - Android app.

[3] P. Bahl and Padmanabhan V.N. Radar: an in-building rf-based user location and tracking system. In 2, pages 775 – 784, March 2000.

[4] Sudarshan S. Chawath. Marker-based localizing for indoor navigation. In Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE, pages 885 – 890, September 2007.

[5] Jaewoo Chung, Matt Donahoe, Chris Schmandt, and et al. Indoor location sensing using geo-magnetism mobisys, 2011.

[6] Sinan Gezici, Tian Zhi, G.B. Giannakis, and et al. Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks. Signal Processing Magazine, IEEE, 22:70–84, July 2005

[7] Google. Android sensor guide.

[8] Kamol Kaemarungsi and Krishnamurthy Prashant. Modeling of indoor positioning systems based on location fingerprinting, 2004.

[9] L. Klingbeil and T. Wark. A wireless sensor network for real-time indoor localization and motion monitoring. In International Conference on Information Processing in Sensor Networks, 2008., pages 39–50, 2008.

[10] Jo Agila Bitsch Link, Felix Gerdsmeier, Paul Smith, and Klaus Wehrle. Indoor navigation on wheels (and on foot) using smartphones. In Proceedings of the 2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN), November 2012.

[11] Android market. Runtastic app.

[12] Oliver Woodman and Robert Harle. Pedestrian localization for indoor environments. In Proceedings of the Tenth International Conference on Ubiquitous Computing (UbiComp 08), pages 362–368, 2008.

[13] Fan Li, Chunshui Zhao, Guanzhong Ding, and et al. A reliable and accurate indoor localization method using phone inertial sensors. In Proceedings of the 2012 ACM Conference on Ubiquitous Computing, pages 421–430, September 2012.

[14] Alessandro Mulloni, Daniel Wagner, Dieter Schmalstieg, and Istvan Barakonyi. Indoor positioning and navigation with camera phones. In *Pervasive Computing, IEEE*, volume 8, pages 22–31, April 2009. 7

[15] MySQL. [Online] Available: https://www.mysql.com [Feb, 1, 2016].

[16] PostgreSQL. [Online] Available: http://www.postgresql.org. Web. 1 Feb. 2016.

[17] SQL Server 2014. [Online] Available: https://www.microsoft.com/en-us/server-cloud/products/sql-server. 3 Feb. 2016.

[18] Oracle. [Online] Available: http://www.oracle.com/index.html. [Feb, 2, 2016].