ON
ATLAS (Smart Library)
a pervasive context aware application

Submitted By:
Nilamkumar Patel
Chintan Shah

Guided By:
Nehal Mehta
Phd Candidate – Dr. Sun’s SCS research group - www.iit.edu/~scs
Professor Xian-He Sun

Computer Science Department
Illinois Institute of Technology
INDEX

Abstract........................................................................................................................ 2
1. Introduction .......................................................................................................... 3
2. Problem Statement ............................................................................................ 4
3. Existing Solution.................................................................................................. 5
4. Solution................................................................................................................... 8
5. Implementation & Result..................................................................................... 17
6. Analysis and Conclusion.................................................................................... 22
7. Reference.................................................................................................................. 25

Encl:
APPENDIX-I Project Proposal
APPENDIX-II Software Requirement Specification
APPENDIX-III Technical Design document
Abstract

In library the searching book is very hectic process, if one doe not have proper direction to reach the rack which held the book. Generally library maintains call number to map relation to get the rack where book is hold. But there are many racks in library and it is very hard to find the particular rack through map. Some good library provides the online map with highlighted rack which contains book. But this is also hard to remember the map as the system is not web based and can run on library’s static PC. In Pervasive computing, context aware development is going on. So by using that technology, we can improve the book searching process. Here, we developed the system smart library, which uses the location aware technology and get the current location of user and give the path to the rack which holds the book. This solution can help tremendous to the library user in searching book.
1. Introduction

1.1 Project Name

ATLAS (Smart Library – a pervasive context aware application)

1.2 Goal

To develop smart Library Environment as the Library is one of the integral parts of any educational environment and used by students as well as faculty members.

1.3 Description

The main purpose of the project is to provide the users a smart library environment, where they can search book online and can be directed with route information to the rack where the book is located. Location awareness has great potential in this application, compared to any other campus tour or city guide application. This is because, in general tour, once user is familiar with the location, in future they might not need this facility often. But in environment such as library, location awareness can be of great help to find books on daily basis. On the other hand the same application can be extended and integrated with many other library resources and make them context aware.

The name ATLAS was given on the basis of the functionalities provided by the system. It is all about concerned with routing the path from the user’s current location to the location of the book in the rack.
2. Problem Statement

All students have accessed library information system to access books. Currently most of the library supports computer based book search engines to find the correct book within fractions of seconds. This project will try to make easy library flow using the pervasive context aware technology.

2.1 Current Scenario

1. Analyzing the current procedure for book search and book checkout -
2. User first uses one of the computers located in the library – to search the required book based on given criteria.
3. The search engine shows the book result in tabular form with Call number of book [usually very long numbers and so very difficult to remember].
4. The user writes down the book call number or may be multiple book number on a paper. The library staff will provide him floor map- with rack.

2.2 Problem in Existing System

This is very time consuming process and sometime if the library is very big, then it will frustrate user very easily.

NOTE: For Detail project proposal see APPENDIX-I
3. Existing Solution

3.1 Survey

We have done survey in below three libraries.

Galvin Library (Main Campus)

Kent Law Library (Downtown Campus)

Chicago Public Library

We used below questionnaire to get requirement and operational feasibility of the project.

Questionnaires

1) To search book, you have to browse book from catalog. Then you have to note down the call no. Then you have to take map of library from library staff and you have to search rack which has that book. Do you have any difficulty in following above procedure of book searching?

☐ Yes ☐ No

If Yes then what kind of change you want?
______________________________________________________________________________________
______________________________________________________________________________________
______________________________________________________________________________________
______________________________________________________________________________________

2) If you are provided solution like..

“You have one small computer or device will show you the path of the book in the rack of the library from your current location in the library or campus.”

According to you, is it good and complete solution?

☐ Yes ☐ No

If No then what kind of solution you think is better?
______________________________________________________________________________________
______________________________________________________________________________________
______________________________________________________________________________________
______________________________________________________________________________________

3) Please tick the thing about which you like to be reminded

☐ Book Pickup

☐ Due Date of Issued book

☐ Some people of your class are in library who may help you in your assignment

☐ Some people of your favourite subject are in Library who wants to share knowledge with you

☐ Other (Specify in below space)
4) What are the other functionalities which you are looking or willing to have in the computerized library system?

Result of Survey

<table>
<thead>
<tr>
<th>Question</th>
<th>Yes</th>
<th>No</th>
<th>Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>95%</td>
<td>5%</td>
<td>Find to get location of book in library</td>
</tr>
<tr>
<td>2</td>
<td>100%</td>
<td>0%</td>
<td>This is enough.</td>
</tr>
</tbody>
</table>

Note: Question 3 and 4 are for the “Ringer” Project.

This all library have system we mentioned in the problem statement..

3.2 Case Study

Project: Library Management system (LMS)

Institute: Nirma Institute of Technology

Ahmedabad, Gujarat, India.

Comparison of feature with ATLAS:

<table>
<thead>
<tr>
<th>Feature</th>
<th>LMS</th>
<th>ATLAS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Book Search</td>
<td>Efficient</td>
<td>Basic(Prototype)</td>
</tr>
<tr>
<td>Map</td>
<td>Good</td>
<td>Good</td>
</tr>
<tr>
<td>Scalability</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>Rack Search on Map</td>
<td>By changing color on the map image</td>
<td>Giving destination Rack Path</td>
</tr>
<tr>
<td>Path</td>
<td>Not Available</td>
<td>Available</td>
</tr>
<tr>
<td>-----------</td>
<td>---------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>Application</td>
<td>Desktop based</td>
<td>Web Based</td>
</tr>
<tr>
<td>User</td>
<td>Static</td>
<td>Dynamic</td>
</tr>
<tr>
<td>Location</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Location</td>
<td>Not Available</td>
<td>Available</td>
</tr>
<tr>
<td>Service</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reusability</td>
<td>None</td>
<td>Location aware and Path finding can be reused for other project also</td>
</tr>
</tbody>
</table>

We did not found any advance library system better than above system so we may be one of the first who are using pervasive computing for library system.

**NOTE:** For Detail Requirement collection and feasibility study see [APPENDIX-II](#)
4. Solution

To help user to do more with doing less, we are proposing smart library environment, where the users can search book online and can be directed with route information to the rack where the book is located. Location awareness has great potential in this application, compared to any other campus tour or city guide application. This is because, in general tour, once user is familiar with the location, in future they might not need this facility often. But in environment such as library, location awareness can be of great help to find books on daily basis. On the other hand the same application can be extended and integrated with many other library resources and make them context aware. The below use case scenario realize the improvement in the process.

4.1 User Scenario and Use-Cases
4.2. User Functional Requirements

<table>
<thead>
<tr>
<th>ID</th>
<th>Description</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>UF1</td>
<td>The system should provide support for searching book through browser-based interface.</td>
<td>1</td>
</tr>
<tr>
<td>UF2</td>
<td>If the book is in the same library where user is, then system should show the path from his current location to the rack where the book is placed.</td>
<td>1</td>
</tr>
<tr>
<td>UF3</td>
<td>If the book is on the different floor, where the user is standing, then the system would show the path of the books from the current location to the location where the book is placed.</td>
<td>2</td>
</tr>
<tr>
<td>UF4</td>
<td>The system should be web based application.</td>
<td>1</td>
</tr>
<tr>
<td>UF5</td>
<td>The system should work on the user’s own devices.</td>
<td>3</td>
</tr>
<tr>
<td>UF6</td>
<td>The system should work at outdoor also (just searching, no path finding).</td>
<td>2</td>
</tr>
<tr>
<td>UF7</td>
<td>It should fetch the content information from Amazon web service.</td>
<td>3</td>
</tr>
<tr>
<td>UF8</td>
<td>User’s account should be maintained properly by the system.</td>
<td>2</td>
</tr>
</tbody>
</table>
4.3. User Non-Functional Requirements

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNF1</td>
<td>Security: The user accounts and the critical information of library should be secure.</td>
<td>3</td>
</tr>
<tr>
<td>UNF2</td>
<td>Availability: The system should be available at any time and any place where internet is available.</td>
<td>1</td>
</tr>
<tr>
<td>UNF3</td>
<td>Efficiency: The path should be shortest</td>
<td>2</td>
</tr>
<tr>
<td>UNF4</td>
<td>Reliability: The location of book should be correct.</td>
<td>1</td>
</tr>
</tbody>
</table>

NOTE: For Detail Requirement Elicitation see APPENDIX-II

4.4. Tools & Platform:

Initially, we started to work with .Net environment but as we started we had resource and incensing problem so we choose to go with open source tools. The reason to choose .Net is to get complete development environment for Database, web service and web based application. But as we had some problems we started with below open source tools which are also same efficient as .Net (individually).

4.4.1. PHP (GD Library)

PHP is good web based application tool which is object oriented tools. Moreover, it is easily wrapped as web services and it has good compatibility with other web services. GD library we chose to support image editing in web based application.

4.4.2. JAVA
As we have JAVA API for EKAHAU Positioning Engine, we also required to make web service in JAVA.

4.4.3. MySQL

MySQL is good & open source Database application, which is also reliable.

4.4.4. Apache Tomcat

To wrap JAVA application as web service we need to have Apache Tomcat web sever.

This is Platform independent as it is web based application.

4.5. Architecture

To implement above requirements, we designed below architecture

This is the over all architecture of the ATLAS.

Subsystems
• Search Engine
• Location Manager
• PathFinder

Database
• Library Database
• Map Image Folder (This folder has map image file and the map db file which consist of nodes of map).

External Component
• EKAHAU Positioning Engine (This is responsible for tracking and giving the current location of the user).

Now, let's discuss design of each component

4.5.1. Library Database Design

This is the design of database from which the search engine will get the data related books.

**E-R Diagrams**
4.5.2. Library Map DB file Design

Library Map DB file is a file which stores the node graph of the map. It stores the nodes of the map with its x and y position and with its connected neighbors’ list.

**Extension:** .db

**Name:** Name should be same as the map image file name. for example if image name id LIBMAP.jpg then the DB file should have name LIBMAP.db

**Format:**

```
1  100 100  2   3  0  0  0  ........  0
2  120 100  3   4  0  0  0  ........  0
...
```

4.5.3. Search Engine Design

This module is responsible for searching the books from user’s keyword. This module searches the books and also searches the rack id and position of rack from the database. Moreover as shown in architecture diagram, this module also call the location manager and pathfinder module to generate the path enabled library map on the browser.

As this module is fully database module. So let’s see Data Flow Diagrams (DFD) of this system.

NOTE: For Detail Design of Search Engine and feasibility study see

APPENDIX-III
4.5.4. Location Manager

Location Manager is a component which is responsible for giving the location of a particular device. This location manager uses the EKAHAU Positioning Engine Server, which tracks the device on a map. So let’s first understand how EKAHAU gets the location.

EKAHAU has three components.

1. Positioning Engine: This engine is always running and tracking the devices which have EKAHAU client installed. It continuously communicates with the client and gets its position after a configured time.
2. Manager: This component allows us to load the map and calibrate it with actual location. This also shows the map and tracked device.
3. Client: The client is installed on the user’s machine. This client continuously sends its position to the Location Engine.

EKAHAU is providing some Java API to get location from the engine. So we can use that API and get the location manager.

Location manager is basically a component which is responsible for talking to the EKAHAU Engine through API and gets the location. As these APIs are in
JAVA, we should implement this location manager in JAVA. Then we wrap it as Web-Service and can be used by any browser client.

We are implementing it as a web service, because the location manager component can be used in other application like Campus Tour and RINGER. So, it will increase the reusability of the developed work.

The below diagram shows how it communicates with EKAHAU engine.

4.5.5. PathFinder Design

We have studied below algorithms.

- A* Algorithm
- Depth First Search
- Krushkal algorithm
- Best First Search

The above all algorithms are basically designed for path finding in Gaming Problem. So they are very much complicated and time consuming. In our problem domain, there is no such complex algorithm required. Because, In our Problem domain, all the path are open. No single node is end point. All are node are connected to each other. So there is no wrong path will be selected. So we modified Best first Search algorithm and made it simple, so it will take less time. Below is our algorithm.

**NOTE: For Detail Design of pathfinder objects see APPENDIX-III**
**Algorithm**

**Procedure findPath(start,end)**

**Input:**
- start: Starting point
- end: Destination Point

**Output:**
- Path: Array of Nodes, ImageFile with path

**Steps:**
1. init = start //Store first node
2. call findP(start,end) and store output in path
3. callDrawPath(path) //which draws the path in file and store it
4. return path

**Procedure findP(start,end)**

**Input:**
- start: Starting point
- end: Destination Point

**Output:** Update path array

**Steps:**
1. Add start in array path
2. if start = end
3. return

- Get the neighbors of start in neighbours array
- i = 0
- flag=0
- for i= 0 to number of neighbors
  - if ith neighbor in InClseList
    - continue
  - flag=1
  - g=Distance(ith neighbor, end)
  - f = g
  - if f is < minimum
    - minimum = f
    - next = ith neighbor
  - if flag=0
    - backtrack and remove start from path
    - next = parent of start
- AddInCloseList(start)
- call findP(next,end)

**Procedure: DrawPath(Path)**

**Input:**
- path array

**Output:** Path drawn image file

**Steps:**
1. x1 = path[0].x //assign fisrt node’s X coordinate
2. y1 = path[0].y // assign fisrt node’s X coridnate
3. loop(steps 4-8) from i = 2nd node of path to end of path
   - x2 = path[i].x
   - y2 = path[i].y
   - Drawline(x1,y1,x2,y2)
   - x1=x2
   - y1=y2
5. Implementation & Result

In Order to implement above design, we have followed the below steps:

5.1. Loading and calibrating the library map in EKAHAU positioning Engine

We load the library map in EKAHAU Positioning engine. Then we calibrated the map by doing site surveying of library. Below screen capture is the site surveyed and calibrated map of library.
5.2. Generating the Library Map DB file.

From above calibrated map, we developed CSV file which has structure mentioned in the Solution section. This file has all the node of above calibrated map.

5.3. Created Sample Database

We created sample database to test the search engine component. This database is same as we mentioned in our solution section. This database is
created in MYSQL below is the screen capture of sample database.

tblcategory:

<table>
<thead>
<tr>
<th>category_id</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAT1</td>
<td>Books</td>
</tr>
<tr>
<td>CAT2</td>
<td>Magazines</td>
</tr>
<tr>
<td>CAT3</td>
<td>Newspapers</td>
</tr>
<tr>
<td>CAT4</td>
<td>Journals</td>
</tr>
<tr>
<td>CAT5</td>
<td>Reports</td>
</tr>
</tbody>
</table>

tbldocument_detail

<table>
<thead>
<tr>
<th>document_id</th>
<th>category_id</th>
<th>name</th>
<th>subject_id</th>
<th>author</th>
<th>call_no</th>
<th>edition</th>
<th>ISSN</th>
<th>magazine_name</th>
<th>frequency</th>
<th>pub_date</th>
<th>no_of_copies</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOC1</td>
<td>CAT1</td>
<td>OPERATING SYSTEM</td>
<td>D1</td>
<td>DISTRIBUTED OF TANBAUM</td>
<td>CAL1</td>
<td>3rd</td>
<td>ISBN 12345-6789</td>
<td>NULL</td>
<td>NULL</td>
<td>2001-01-05</td>
<td>5</td>
</tr>
<tr>
<td>DOC2</td>
<td>CAT1</td>
<td>SOFTWARE METRICS</td>
<td>SM</td>
<td>SOFTWARE MET FLEEGER</td>
<td>CAL2</td>
<td>2nd</td>
<td>ISBN 23456-7890</td>
<td>NULL</td>
<td>NULL</td>
<td>2002-02-06</td>
<td>6</td>
</tr>
<tr>
<td>DOC3</td>
<td>CAT1</td>
<td>DATABASE MANAGEMENT</td>
<td>ADG</td>
<td>DATABASE MANI SCHLERTZ</td>
<td>CAL3</td>
<td>3rd</td>
<td>ISBN 34567-8901</td>
<td>NULL</td>
<td>NULL</td>
<td>2002-06-02</td>
<td>4</td>
</tr>
<tr>
<td>DOC4</td>
<td>CAT1</td>
<td>SOFTWARE ENGINEERING</td>
<td>SE</td>
<td>SOFTWARE ENG PRESSMAN</td>
<td>CAL4</td>
<td>3rd</td>
<td>ISBN 45678-9012</td>
<td>NULL</td>
<td>NULL</td>
<td>2003-02-23</td>
<td>9</td>
</tr>
<tr>
<td>DOC5</td>
<td>CAT1</td>
<td>OBJECT ORIENTED PROG</td>
<td>DOP</td>
<td>OBJECT ORIENT KORTH</td>
<td>CAL5</td>
<td>5th</td>
<td>ISBN 56789-0123</td>
<td>NULL</td>
<td>NULL</td>
<td>2005-02-10</td>
<td>4</td>
</tr>
</tbody>
</table>

tbldocumentstatus

<table>
<thead>
<tr>
<th>status_id</th>
<th>document_id</th>
<th>rack_id</th>
<th>status</th>
<th>status_upd_date</th>
<th>comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>STA1</td>
<td>DOC1</td>
<td>1A</td>
<td>AVAILABLE</td>
<td>2004-12-10</td>
<td>AVAILABLE</td>
</tr>
<tr>
<td>STA2</td>
<td>DOC2</td>
<td>2A</td>
<td>AVAILABLE</td>
<td>2005-10-22</td>
<td>AVAILABLE</td>
</tr>
<tr>
<td>STA3</td>
<td>DOC3</td>
<td>3A</td>
<td>AVAILABLE</td>
<td>2002-12-03</td>
<td>AVAILABLE</td>
</tr>
<tr>
<td>STA4</td>
<td>DOC4</td>
<td>4A</td>
<td>AVAILABLE</td>
<td>2003-12-05</td>
<td>AVAILABLE</td>
</tr>
<tr>
<td>STA5</td>
<td>DOC5</td>
<td>1A</td>
<td>AVAILABLE</td>
<td>2005-03-12</td>
<td>AVAILABLE</td>
</tr>
</tbody>
</table>

tblrack_location

<table>
<thead>
<tr>
<th>rack_id</th>
<th>floor_id</th>
<th>capacity</th>
<th>availability</th>
<th>upper_rack_no</th>
<th>lower_rack_no</th>
<th>rack_GPS_x1</th>
<th>rack_GPS_y1</th>
<th>rack_GPS_x2</th>
<th>rack_GPS_y2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>25</td>
<td>Y</td>
<td>1</td>
<td>2</td>
<td>100</td>
<td>200</td>
<td>100</td>
<td>200</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>25</td>
<td>Y</td>
<td>1</td>
<td>3</td>
<td>200</td>
<td>300</td>
<td>200</td>
<td>300</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>25</td>
<td>Y</td>
<td>3</td>
<td>4</td>
<td>300</td>
<td>400</td>
<td>300</td>
<td>400</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>25</td>
<td>Y</td>
<td>4</td>
<td>5</td>
<td>400</td>
<td>500</td>
<td>400</td>
<td>500</td>
</tr>
</tbody>
</table>

tbl_amazon_docdb

<table>
<thead>
<tr>
<th>document_name</th>
<th>index_file</th>
<th>sample_pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

5.4. Setting up the environment for server

We installed Apache server on windows machine and PHP with GD library. This machine will use the web server. And also second machine with EKAHAU server,
which will track the devices and give the location.

5.5. Implementing Path finder, Search Engine and Location Manager

We implemented above mentioned modules. The result of above application is as shown in below screen captures.

**Screen1**: In first screen user searches for the keyword “Software”.

**Screen 2**: In Second screen Search engine has displayed the results matching with “software” keyword.

**Screen3**: In third screen User has shown the path of clicked book on library map.
<table>
<thead>
<tr>
<th>Book Name</th>
<th>Title</th>
<th>Author</th>
<th>Edition</th>
<th>ISBN</th>
<th>Publication Date</th>
<th># of Copies</th>
</tr>
</thead>
</table>
6. Analysis and Conclusion

6.1 Analysis

Challenges:

1. **Tool Challenge:** We have created Search engine with PHP and Location Manager in JAVA. As we can not make JAVA and PHP bridge we have implemented Location Manager as web service.

   **Solution:**

   If we can make bridge between JAVA and PHP then there would not be need to create web-service. (By the way, creating web service has many other advantages).

2. **Algorithm Challenge:** we started to work on developing the algorithm on
our own, but we found that it was not much reliable for the Path Finding.

**Solution:**

If we have implemented the well-known algorithm like Best First Search and A* algorithm then reliability would be more than current project.

3. **EKAHAU Challenge:** EKAHAU is over all good positioning engine, but sometime it fails to give very much accurate results. So the accuracy is some less by using it.

**Solution:**

We can make same positioning engine and instead of taking service from EKAHAU we would be having our own accurate Engine which has a good reliability.

**Future Enhancement:**

1. Instead of just showing path we can update the current position of the User while he is walking towards rack.

2. We will be having good algorithm which has more accurate result then current one.

3. We can wrap the Path finder and web service and make it usable by other campus’s location aware project.

4. We can have a better UI of MAP on browser with zoom in and zoom out facility.

5. This library application is for one floor one building only. In future we can make it multiple building and multiple floors supported.

**6.2 Conclusion**

An ATLAS is pervasive computing application software, which is providing the Path Finding Facility to the Library Users in such a manner that they can see the location of the books, magazines, journals, etc.... This project has a good impact on starting
use of pervasive computing application in library environment. This project has somewhat less accuracy in location and path finding aspect, but the approach of hitting the target was good.
7. Reference


3. Hawk tour - context aware tour guide - www.hawktour.net


5. PHP - http://www.php.net/

6. php class repository - http://www.phpclasses.org/


