Low Cost Interactive Area of Interest Locator

Annie Hii Toderici
San Jose State University
California, USA

SVG Open 2010
Overview

• Brief history of maps
• Current map locator used
• Proposed alternative
• Benefits of the proposed alternative
• Demo
• Future development
• Q&A
History

- Global Positioning Systems (GPS) Navigation devices are interactive map systems for the roads.
- Static maps (paper, posters) are available in libraries, malls, stores, schools, etc.
- Such entities usually cannot afford building interactive maps due to high cost of licensing technology.
Typical Library Mapping System

- The current map is a static GIF file (others may use JPEG). Raster format with no zoom in capability.
- Only provide a general area of an item. Example: “1st floor – Science Fiction area” is marked on the map with no further subdivision, due to lack of space (maps must be displayed on screen).
- Locating for an item thus will be imprecise and time consuming.
Proposed SVG Map
Locating Items: Previous Work

• Radio Frequency Identification (RFID) based tracking of items. Most precise.
• Infrared Data Association Communication modules (IrDA) with Light Emitting Diode (LED). Still needs manual scanning.
• GPS (for big items!)
• Bottom line: Not cost efficient!
Current Solution

• Since libraries cannot afford such expensive systems, they employ barcode scanners or RFID chips (for the wealthier areas).
• Barcodes (or RFIDs) encode the “item ID”.
• The library has a database of Item ID to aisle mappings.
Development and Implementation

• Make use of existing unique barcode system
• Intergraded existing map into SVG formatted map
• Create an XML-based database
• Implemented with JavaScript to read input and return the data
Converting the GIF map into SVG

- **Using Vector Magic** (commercial software)
  - Provide 2 free conversions
  - Simply upload the file into the site, it returns the SVG file without user interaction
  - The result still needs to be processed in Inkscape
- **Using Inkscape** (open source)
  - A “free” (as in “free speech”, and as in “free beer”) application
  - Open the file and then select “Path and Trace Bitmap” to generate SVG map
  - Remove markings and add rows of bookshelves
  - **Alternative:** start drawing the map from scratch.
Data Integration

- DTD for the XML storage of the books’ information:

```xml
<?xml version="1.0" standalone="yes"?>
<!DOCTYPE books [ 
<!ELEMENT books (floor*)>
<!ELEMENT floor (book*)>
<!ELEMENT book (range)>
<!ELEMENT range (start, end, aisle)>
<!ELEMENT start (#PCDATA)>
<!ELEMENT end (#PCDATA)>
<!ELEMENT aisle (#PCDATA)>
<!ATTLIST floor id ID #REQUIRED>
]>
```
XML Database

```xml
<books>
  <floor id="1">
    <book>
      <range>
        <start>J10</start>
        <end>J90</end>
        <aisle>1</aisle>
      </range>
      <floor>1</floor>
    </book>
  </floor>
  <floor id="2">
    <book>
      <range>
        <start>Q11</start>
        <end>Q22</end>
        <aisle>1</aisle>
      </range>
      <floor>2</floor>
    </book>
  </floor>
</books>
```
Implementation: Chrome Extension

• Rationale:
  – Easy way to lookup the location of an item without needing to copy and paste the item ID into a new window.
  – This browser supports SVG natively.
  – Designed specifically for the Mountain View Public Library search system.
  – Allows the search data to be sent to a PHP script, then to return the map with the shelf highlighted.
Use-Case

• The user navigates to the Mountain View Library Website.
• Once a search for a book is submitted, it returns the call ID for that book.
• The Book Finder extension become active once it detects call IDs within the HTML.
• The user clicks on the icon on the address bar to open up a map with the aisle highlighted.
PHP Backend / JavaScript Frontend

- Called by the Chrome extension.
- Displays the SVG map.
- JavaScript code calls a PHP lookup script to get the aisle ID for the current item.
- If a valid ID was found, JavaScript code processes the SVG DOM and highlights the aisle of interest.
- XMLHttpRequest is used to communicate between the frontend and backend.
XML Responses

- No error, then it will return the map
- Item not found, then it will return an error

Example 1: The queried book was found in aisle with the id "aisle50":

```xml
<response>
  <error>0</error>
  <location>aisle50</location>
</response>
```

Example 2: The queried book was not found in the database:

```xml
<response>
  <error>1</error>
  <message>Item Not Found!</message>
</response>
```
Prototype Implementation
Benefits of the proposed system

- Uses existing maps and the existing barcode system with no need for extra hardware.
- Creating SVG maps is easy (and converting to SVG can be done inexpensively).
- A simple XML database maps the item range ID to an area ID (this can be integrate from the existing system).
- Once the code is finished, they can be reused for all other libraries with minimal changes (mostly for interfacing with the existing software).
- Low cost:
  - Minimal effort needed.
  - No maintenance unless the library map changes.
Demo
Future Development

• Improve the user interface from prototype to product quality.
• Get all the libraries to adapt this simple and cost effective mapping system!
Questions?

• Please email your questions or suggestions to anniehii@gmail.com

• Thank you