Bubble's Network – Prefuse as a visualization kit

Denzil Sócrates Teixeira Ferreira
University of Madeira
SAUI – Prof. Vassilis Kostakos
a2008301@max.uma.pt
+351 91 713 25 92

Maria do Carmo Oliveira de Freitas
University of Madeira
SAUI – Prof. Vassilis Kostakos
a2014704@max.uma.pt
+351 91 836 61 91

ABSTRACT
In this paper we describe our proposal to visualize a social network. Using the existing Prefuse visualization toolkit for Java, data visualization is achieved by analyzing active Bluetooth devices around a Bluetooth scanner. Cityware framework was changed to store all the incoming detections data on to a GraphML file and sends them to Cityware for a more profound analysis.

Author Keywords
Cityware, prefuse, bubble view, graphml database.

ACM Classification Keywords
H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION
The Cityware framework studies and analyses social networks online and offline. It uses Facebook for spreading the word on how your social network can go beyond the internet realm, reaching out to the real world. By itself, Cityware already stores devices connections, but there aren't any ways to visualize this data. So, as requested, we imagined a way to visualize the connections between users, by providing a different approach. No boring graphs or tables, but something new, people become bubbles on an endless ocean.

DEVELOPMENT TOOLS
The interface has been constructed using Java Swing framework, on NetBeans 6.0.1 IDE for MacOSX and Ubuntu 7.10 Linux. In each case, the interface is rendered as expected, maintaining the look and feel of each operating system.

CITYWARE
Cityware stores the information that a device sends by uniquely identifying the device by its Bluetooth ID and also a given password which is assigned by the Cityware application online. In order to make our visualization as user friendly as we can, we provide the means for the user storing its own password on the server, so that it does not have to rewrite it again and again. The user can remove its stored password by unchecking the remember checkbox.

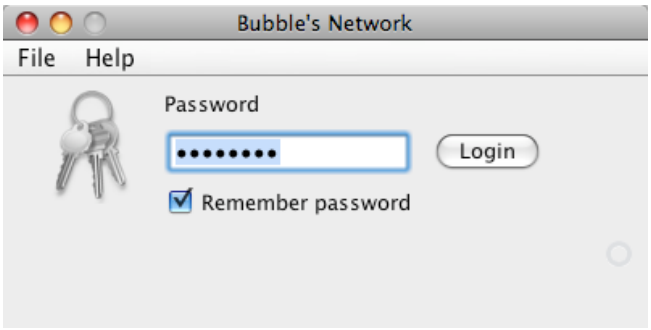

Figure 1: Login form presented to the user

The code of Cityware server node has been extended to support the collection of data in GraphML format. This represents data on a graph, connecting nodes (which represent people) by edges (relationships).

On our approach, we store people as followed:

```
<node id="0">
    <data key="image">bubble.png</data>
    <data key="timelast">1205887576385L</data>
    <data key="address">001E52D20FC0</data>
    <data key="name">Denzil Ferreira&#8217;s
MacBook</data>
    <data key="visits">1L</data>
    <data key="timesum">0L</data>
</node>
```

The data key store the attributes of the nodes. The image attribute can be "bubble.png" or "bubble_off.png", depending if the device is online or offline. Timelast is the date and time in which the device appeared in range of the scanner. Address stores the unique ID of the Bluetooth device, used in the update of the status and avoidance of data redundancy. Name stores the friendly name of the device, this is set by the Bluetooth owner. Visits stores the number of times the device has been away and then has come back in range. The timesum stores the amount of time the device has been visible to the scanner.

KEEPING THE EYE ON THE BALL
Following the course guidelines, we provided a way of monitoring the status of the network scanner, not by displaying distracting elements or sounds but by a command line output, visible on the server status window.

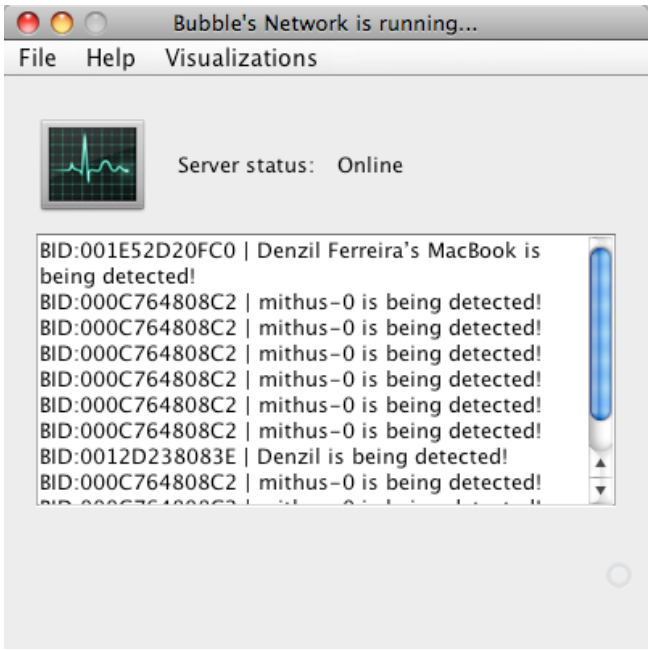

Figure 2: Server monitoring window

VISUAL SOCIAL NETWORK

A picture speaks more than a thousand words. Bubbles on a blue ocean, gives an overview of what's going on the surrounding area.

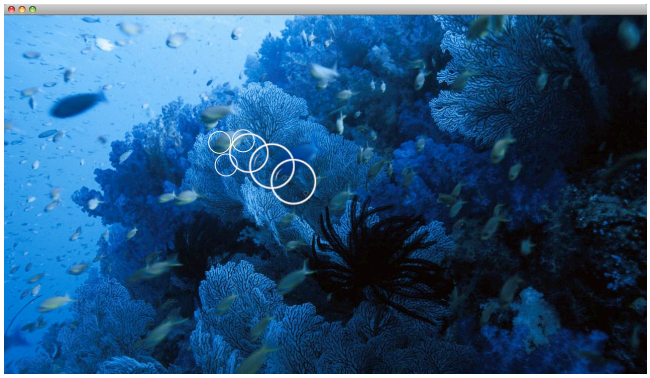

Figure 3: Bubble's network visualization

RELATIONSHIPS

Time is uncontrollable. Hence it never stops growing, and so does our visualization. Users are related by the amount of time they spent together. If two users appear online in a time span of 5 minutes between them, it means that in a way they share the same space at the same time, so they are close to each other.

HIERARCHY

Societies are often hierarchical. We represent the hierarchy of the persons involved by sizing them differently. This takes in account the number of times the device as came close to the device, becoming bigger as the number of times it appears increases.

STATUS

Everyone likes attention. We notice someone that's active if it stands out in the middle of a crowd. In our visualization, the devices that are active on the past 5 minutes become brighter than the ones that are away for more than 5 minutes, becoming a fading bubble. In order for them to stand out again, they just have to come back online and eventually they become bigger.

CONCLUSION

After two weeks work, exploring and trying to understand Prefuse visualization framework, we came to a solution that presents social networks on a different manner. By using this approach, what used to be boring tables, becomes a pleasing and easy way to follow the growth of the network. More information about the bubbles can be acquired by hovering the mouse over the bubbles, giving the name of the device detected. We wanted to keep the visualization the cleanest possible, using a blue ocean background that highlights the white bubbles and are relaxing to the eyes, but yet keeping informative, and possible to identify user groups and importance of the elements.

ACKNOWLEDGMENTS

We'd like to thank Emanuel Fernandes, Tiago Camacho for the patience on turning on and off their devices countless times during this project test runs.

REFERENCES

1. ACM Computing Classification System: H.5. http://www.acm.org/class/1998/H.5.html.
2. CS 4767 :: Prefuse Tutorial.

   http://www.cs.mun.ca/~hoeber/teaching/cs4767/notes/04-prefuse/
3. Defining and Starting a Thread (Java Tutorials) http://java.sun.com/docs/books/tutorial/essential/concurrency/runthread.html.
4. GraphML Primer

   http://graphml.graphdrawing.org/primer/graphml-primer.html
5. GraphML Specification

   http://graphml.graphdrawing.org/specification.html.
6. Infovis-Wiki – Prefuse tutorial. http://www.infovis-wiki.net/images/d/d7/Prefuse-tutorial-20061127-handout.pdf.
7. Programming Workshop – Creating a simple graph visualization http://cs.marlboro.edu/courses/spring2006/programming_workshop/wiki/Creating_a_simple_Graph_Visualization.