Opening Pervasive Computing to the Masses Using the SEAP Middleware

PerWare 2009 – PerCom 2009

Seth Holloway, Drew Stovall, Jorge Lara-Garduno, and Christine Julien
Mobile and Pervasive Computing Group
The University of Texas at Austin
March 9, 2009
Pervasive Computing Middleware

- We all want to make pervasive computing a reality
- Middleware will make this happen
- But how?
How to Proceed

- Focus on users
- The Internet grew when users created their own content and behaviors
- Pervasive computing can follow suit

Growth of Internet

http://mpc.ece.utexas.edu/SEAP/
Seth Holloway PerWare 2009
Sensor Enablement for the Average Programmer

- There are a lot of *average programmers* who could be programming pervasive computing applications using

  - *Devices*
    - *Sensors* – take readings
    - *Actuators* – perform commands
An Example Aware Home

A user has an aware home with several specialized sensing devices. It was installed by a professional using a proprietary system.
An Example Aware Home

The user wants to add a new sensor; however, he cannot integrate it himself and it will cost hundreds to get it installed by a professional.
Challenge

- Today’s sensors and actuators use a wide variety of languages, data models, and communication protocols
- Creating custom component interactions is challenging, often requiring new specializations
Aware Home

- The user understands the behavior of the existing system

Light Sensor → Behavior → Lights On/Off
Aware Home

But the user cannot integrate new components

Light Sensor

Behavior

RFID

Lights On/Off
Aware Home

- Behavior is in a black box

Light Sensor → Behavior → Lights On/Off

RFID
Web Applications

- Widely understood
Aware Home Revisited

- Open behavior
- Allow devices to speak “web”
New Aware Home

- Device integration is simplified
Solution

- Use HTTP as a common data model and communications protocol

- Sensors “post” data to forms, actuators “get” commands from pages

- **End-user programmers** construct applications using their **favorite programming language** and its interface to HTTP
SEAP Conceptual Architecture

[Diagram showing the SEAP architecture with Sensor, Host, Application Server, User Code, Actuator, and HTTP and Control Flow connections]
Example Application: Spot-to-Bot

- Use a Sun SPOT to drive a Roomba
Spot-to-Bot Without SEAP

Spot-to-Bot without SEAP requires:

• Sun SPOT variant of Java ME
• Sun SPOT build tools
• Java SE
• Java Player client library
• Linux system administration to build and install Player server
Spot-to-Bot With SEAP

Spot-to-Bot with SEAP requires:
• Your favorite programming language
• Any web application framework

http://mpc.ece.utexas.edu/SEAP/
Seth Holloway PerWare 2009
SEAP Spot-to-Bot

Sensors

User Application

Actuators

http://mpc.ece.utexas.edu/SEAP/
Seth Holloway PerWare 2009
Concrete Implementation (Sensor)

HTTP Post

Rendered from

```html
<jsp:useBean scope="application" id="spot3F5D"/>
<jsp:setProperty name="spot3F5D" property="x"/>
<jsp:setProperty name="spot3F5D" property="y"/>
<jsp:setProperty name="spot3F5D" property="z"/>
<html>
<head>
    <title>Accelerometer Data for Sun SPOT 0000.3F5D</title>
</head>
<body>
    <form action="accelerometer.jsp" method="post">
        X: <input name="x" value="\$spot3F5D.getX() \$">
        Y: <input name="y" value="\$spot3F5D.getY() \$">
        Z: <input name="z" value="\$spot3F5D.getZ() \$">
        <input type="submit"/>
    </form>
</body>
</html>
```
Concrete Implementation (Actuator)

HTTP Get

0.5

Rendered from

```jsp:useBean scope="application" id="tiltometer"
<%= (tiltometer.getY()+1.0)/2.0 %>`
SEAP Spot-to-Bot

- This application exists in our lab
SEAP Spot-to-Bot Extension

- Roomba replaced with pan-tilt assembly
- Same code as Spot-to-Bot
SEAP Spot-to-Bot Extension

- Added RFID to control access
  1) New HTML form
  2) If statement in logic
Benefits of SEAP

- Web programming is approachable to average programmers
- Application frameworks are available in a variety of programming languages
- Support from existing HTTP and web infrastructure, products, and knowledge
- Easy integration with existing web paradigms and tools
Drawbacks of SEAP

- Some devices cannot form HTTP
  - Use sensor protocols to network perimeter then turn into HTTP

- Communication overhead
  - Traffic
  - Connection set-up
  - Data parsing
  - SEAP is not for high performance applications, it is for enabling applications
Making SEAP a Reality

- We have several working prototypes
- Sensor manufacturers can easily provide the necessary code
  - Devices create HTTP requests individually or through base station
- Users can quickly create new pervasive computing applications
- Users can easily augment existing applications to use sensors and actuators
Future Directions

- Expand research into end-user programming, HCI, web (REST), hosted solutions with meta languages
- Encourage high school students and college undergraduates to develop applications using SEAP
- Create configuration utility for rapid deployment of sensors
Conclusions

- Sensor Enablement for the Average Programmer (SEAP)
- Data transportation middleware
- Unify sensor data using existing technologies
- Focus on usability
- Make pervasive computing a reality
Thank you for your time and attention

Seth Holloway
sethh@mail.utexas.edu
http://mpc.ece.utexas.edu/SEAP/
Citation Information

@InProceedings{,
  author = {Seth Holloway and Drew Stovall and Jorge Lara-Garduno and Christine Julien},
  title = {Opening Pervasive Computing to the Masses Using the SEAP Middleware},
  year = {2009},
  booktitle = {Proceedings of the Middleware Support for Pervasive Computing Workshop},
  address = {Galveston, Texas},
  month = {9--13 March},
  notes = {PerWare09}
}

http://mpc.ece.utexas.edu/SEAP/ Seth Holloway PerWare 2009