A Magic Lens for Revealing Device Interactions in Smart Environments

Simon Mayer, Yassin N. Hassan, Gábor Sörös

ETH Zurich, Switzerland
A thousand interacting devices at home...

...humans should stay in control!
A magic lens for revealing these interactions
Overview

1. Logging Communication Traces
   - Record application-layer (HTTP) communication
   - Collect and causally order messages

2. Web Visualization
   - Network topology
   - Timeline view and graph view

3. Magic Lens Visualization
   - Recognize (multiple) devices
   - AR View

Logging Agents → captured interactions

1. Logging
   - REST API
   - Inspect
   - Web Sockets
   - (Storage)

2. Web Visualization
   - instant updates

3. Magic Lens Visualization
   - user interface
1. Logging Communication Traces

2. Web Visualization

3. Magic Lens Visualization
Web of Things

- **Internet of Things**: Focus on **Connectivity**

- **Web of Things**: Focus on **Interoperability**
  - “Everything is a Web Resource”
  - Inherit **desirable properties** of the Web: scalability, caching, addressing, security, etc.
  - **Simpler usage** (and debugging) for humans: the Web browser
  - **Simpler interaction** with other devices: Internet Media Types and Content Negotiation

- In this paper: Every device runs an embedded Web server
  - Grizzly server, based on the Java Non-Blocking IO
  - REST APIs: Idealized Web Architecture
Recording Device Interactions

Our UbiComp2013 paper

Non-Intrusive

Intrusive
HTTP packet inspection

- **Non-intrusive** sniffing on router: Information on the network layer
  - Only IP-addresses and headers
  - Or even restricted to layer 2 frames (L2-Switching for WLAN-WLAN)

- **Intrusive** logging on clients: **Full application-layer information**
  - Content + Media Type (e.g., display transmitted images)
  - Inspect encrypted traffic (we log after the decryption step)
  - Use HTTP to piggy-back management information (e.g., vector clocks)
  - Full URLs: Ability to distinguish devices behind a translation gateway and differentiate HTTP endpoints (e.g., robot.org/motor2 vs. robot.org/sensor4)

- Drawback: Install the logging agent on every device
Logging Agents

- Java Agents **modify the bytecode** of the classes loaded by the JVM at runtime without changing its source code
  
  ```
  java -javaagent:LoggingClient.jar -jar WoTDevice.jar
  ```

- Adds logging code to the Java classes `URLConnection` and `HttpServer`

- Information about recorded packets is sent to the logging server
- The server can be any connected device with enough computing resources
Logging Server

- Logging agents report through a REST API (1)
- Updates are pushed to subscribers using HTML5 WebSockets (2)
- History can be queried through another REST API (3)

- The server orders incoming messages causally and identifies interaction chains.
Causal ordering: messages

An interaction chain is a set of interactions in which all requests and responses are the consequence of an initial HTTP request.

Send/Receive events are logged and reported to the server.
Causal ordering: problem

Reports may arrive in a causally incorrect order. The server must reconstruct the causality chain!
Causal ordering: vector clocks

Causal ordering: implementation

HTTP header information added to outgoing request:

```
GET /volume HTTP/1.1
Host: 192.168.1.147
X-Clock-192.168.1.147_9001: 1
```

HTTP header information added to the response:

```
HTTP/1.1 200 OK
Content-Type: text/html
X-Clock-192.168.1.147_9002: 4
X-Clock-192.168.1.147_9001: 7
Content-Length: 0
```
Aggregating interaction chains

HTTP header information added to outgoing requests:

```
GET /volume HTTP/1.1
Host: 192.168.1.147
X-Interaction: e989eaa6-2689-4c06-9828-bdda2c4eaf23
```
1. Logging Communication Traces

2. Web Visualization

3. Magic Lens Visualization
Web Visualization – Graph view
Web Visualization – Timeline View
1. Logging Communication Traces
2. Web Visualization
3. Magic Lens Visualization
Recognizing Smart Devices

first prototype (non-intrusive sniffing)

AR markers for tracking
QR codes for encoding URLs

current system (intrusive logging)

Visual features, no markers required
URLs registered manually
Object Recognition

- **ORB Feature Detector and Descriptor**
  - Oriented FAST (Features from Accelerated Segment Test)
  - Rotated BRIEF (Binary Robust Independent Elementary Features)
  - [Rosten2005] [Calonder 2010] [Rublee 2011]

- **Bag of Words (binary version) histograms** [Grana 2013]

- **One binary SVM per class** (10 object classes + noise class)
Input: 15-30 snapshots of each device

The method is fast, but limited to the trained objects
Requires re-training when a new class is added
What is a representative noise class?
Multiple Object Detection

- FAST feature detector
- DBSCAN for spatial clustering
  
  **Density-Based Spatial Clustering of Applications with Noise**
  
  number of spatial clusters is not known in advance (as opposed to k-means)

- Kalman filter for object centroid tracking
- Low-pass filter on each class label
The magic lens in action

video
Smart electricity meter and smart car

simulated communication
no Java VM on the smart meter

real communication
Java VMs in CloudThink

www.bitstoenergy.ch

www.cloud-think.com
Limitations

- Recognition is relatively fast, but limited to the trained objects. **Need to re-train** if a new device is added.

- Segmentation requires feature-rich objects and little background clutter. Use (pointing) **gestures** to select devices?

- Scaling with the number of devices. Use **context** to limit search space (devices in *this* room).

- Logging client uses Java Agents. **Java-based** smart devices only.
Summary

Today, our tool
- ...recognizes smart devices based on their visual features
- ...shows their interactions in a magic lens view
- ...allows users to intuitively monitor information flows
- ...gives more control over the smart home network

Future work should
- ...integrate better object recognition techniques
- ...add networking rules by a flick of a finger
- ...categorize communication (e.g., "Post image to Facebook", "Download song")
Thank You

谢谢

Credits
Markus Schalch
Bram Scheidegger
Claude Barthels
Marian George
Christian Beckel
References

- Rosten - Machine Learning for High-Speed Corner Detection (ECCV 2006)
- Calonder - BRIEF, Binary Robust Independent Elementray Features (ECCV 2010)
- Rublee - ORB, an efficient alternative to SIFT or SURF (ICCV 2011)
- Grana - A Fast Approach for Integrating ORB Descriptors in the Bag of Words Model (EI 2013)
- Caetano - Representing Local Binary Descriptors with BossaNova for Visual Recognition (SAC 2014)
- Mayer - Uncovering device whispers in smart homes (MUM 2012)
- Mayer - Device recognition for intuitive interaction with the web of things (UbiComp 2013)
- Guinard - From the Internet of Things to the Web of Things (IoT 2011)
- Csurka – Incorporating Geometry Information with Weak Classifiers for Improved Generic Visual Categorization (ICIAP 2005)
- Mattern - Virtual time and global states of distributed systems (WPDA 1988)
Image Sources

- http://andrewbleakley.com
- http://www.webmarchand.com
- http://www.patentspostgrant.com/
- http://www.orkin.com
- http://sweetclipart.com
- http://la-matrice.org
- http://learn.randolph.k12.ma.us
- http://theparisreview.org
- http://bestclipartblog.com
- http://freepik.com
- http://www.productdose.com
- http://westwoodblinds.co.uk
- http://www.ambientdevices.com
- http://www.pats.ua.ac.be
- http://www.vesternet.com
- http://www.rovingnetworks.com
- http://www.qualitymechanicalservices.com
- http://chantalteakettle.org
- http://blog.eeye.com
- http://cartographersguild.com
- http://haverford.edu
- http://clker.com
- http://academycomicalarts.blogspot.com
- http://www.websenor.com