BOSS: Building Operating System Services

Stephen Dawson-Haggerty, Andrew Krioukov, Jay Taneja, Sagar Karandikar, Gabe Fierro, Nikita Kitaev, and David Culler
BOSS

- A collection of services forming a distributed operating system
  - Scalable
  - Portable
Existing Building Systems

• Extensive internal systems to manufacture an indoor environment
  – Heating, safety, connectivity
• Many different vendors
Motivating Applications

- HVAC Optimization
  - Dynamically adjust the volume of ventilation air
- Personalized Control
- Energy Audit
# Design

<table>
<thead>
<tr>
<th>Architectural component</th>
<th>Functional requirements</th>
<th>Placement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware presentation layer</td>
<td>Expose the primitive low-level operations of hardware using a common interface.</td>
<td>Distributed as close to the physical sensors as possible (ideally, co-located).</td>
</tr>
<tr>
<td>Control transaction manager</td>
<td>Provide “all or nothing” semantics when applying control inputs; provide rollback of actions on failure, cancellation, or expiration.</td>
<td>Within the same failure domain as the HPL used to affect the changes.</td>
</tr>
<tr>
<td>Hardware abstraction layer</td>
<td>Map the low-level functions of the physical hardware to higher-level abstractions.</td>
<td>Anywhere.</td>
</tr>
<tr>
<td>Time series service</td>
<td>Maintain a history of readings from the sensors and actuators; provide application interface to data.</td>
<td>Replicated; may be offsite.</td>
</tr>
<tr>
<td>Authorization service</td>
<td>Approve application requests for access to building resources.</td>
<td>Anywhere.</td>
</tr>
<tr>
<td>Control processes</td>
<td>“User processes” implementing custom control logic.</td>
<td>Anywhere.</td>
</tr>
</tbody>
</table>
Hardware Presentation Layer (HPL)

- Present hardware capabilities through uniform, self-describing protocol
- Maps a sensor or actuator into a point, which produce time series.
- Services for each point: read/write, subscribe to changes or receive notifications, or retrieve/append metadata
Hardware Presentation Layer

- Naming
- Metadata
- Buffering and Leasing
- Discovery and Aggregation
Hardware Abstraction Layer

- Approximate query language
- Abstracts logic to control building components using drivers
- Control sharing at the level of functional component
Time Series Service

- **Stream selection language**
  - Inspect/retrieve metadata about time series
- **Data transformation language**
  - Apply a pipeline of operators to the retrieved data to perform data cleaning operations
Control Transactions

• Set of actions to be taken at a particular time
• Lease time
  – When actions are valid
• Revert sequence
• Error policy
Control Transactions

- Point-level operation
  - Priority level
  - Locking Strategy
- Priority array
  - Based off BACnet
  - Written to a point based on priority level
Authorization Service

• Two-stage approve/verify process
  – Application registers intent
  – Automatic approval or sent to building manager

• Verify access permissions using online server
  – Places authorization service on critical path of all application actions
Control Processes

• Larger, complex long-lived blocks of logic
• Transaction manager handles failures
  – Revert to lower priority CP
Implementation and Evaluation

• Hardware Presentation
  - HPL is revised version of sMAP
    • RESTful access to data sources and actuators.
• Implemented 25 driver modules to integrate with three BMS vendors
Hardware Abstraction Layer

- Semantic Query Language
Hardware Abstraction Layer

- Drivers
  - Bound to HPL points based on metadata provided by HPL
Time Series Service

- Stores, selects, and cleans real-time and historical data
- Readingdb
  - Custom algorithm, others not suitable for large-scale purposes
Time Series Service

• Data Selection
  – Each data stream gets a unique identifier

• Data Transformation
  – Numpy library
Transaction Manager

(a) Unstable behavior around a transition

(b) Specialized reversion sequences can deal with this problem
Applications

- HVAC Optimization
- Personalized Control
- Energy Audit
HVAC Optimization

- Traditionally separate control loops
  - Fresh air intake
  - Per-room airflow
  - Detecting occupants
- Coordinated control across normally independent building components
Personal Control

- Blast air into space in response to user request
- Give user request higher priority
- HVAC optimization regains control when application finishes
- In event of failure, revert changes
Auditing and Baselining

- HPL and time series service to read historical data from all meters
- Train baseline model of power consumption
- 17% decrease in power consumption
Energy Savings

Savings: 17.06%
27.99kW
Related Work

• Ubiquitous Computing
  – Many home systems
  – BOSS allows applications to be written in terms of components' relationship with other components
Related Work

- **Metadata**
  - Project Haystack – uses list of tags and rules to describe building components

- **Protocols**
Future Work

- Inferring HAL with minimum effort
- Software Defined networks
  - Control intent defined abstractly, SDNs could be used to enforce access control
Discussion

- Other uses other than HVAC
- More data on evaluation