Brick: Towards a Unified Metadata Schema For Buildings

http://brickschema.org/
A Vision for “Smart Buildings”

**Infrastructure:**
- Sensors
- Equipment

**Management Services:**
- APIs
- Data Storage
- Monitoring
- Search
- Access Control
- Privacy

**Buildings:**
- Home
- Office
- Factory
- Research Lab
- Hospital
- Shopping Mall

**Applications:**
- Demand Response
- Occupant Interaction
- NILM
- Occupancy Models
- Predictive Control
- Fault Detection
- APIs
- Data Storage
- Access Control
- Motion Sensor

**Occupant Interaction**

**Predictive Control**

**Fault Detection**

**NILM**

**Demand Response**

**Occupancy Models**

**APIs**

**Data Storage**

**Access Control**

**Motion Sensor**

**Fire Safety**

**Thermostat**

**HVAC**

**Lighting**
Data From Buildings Faces Significant Challenges in Integration

- Apps need to understand building metadata
  - Location of sensors and equipment
  - How equipment connect with each other
  - Configuration parameters of control systems

- Temperature sensor as an example
  - What? -> air, water
  - Where? -> room, exhaust, refrigerator
  - How is it used? -> control system, fault

- But metadata not consistent or machine readable

- Metadata varies with building, vendor, type of system
Metadata For An Existing Building

Source: EBU3B building at UCSD, built in 2004. Word cloud limited to 200 most frequent words
We Need a Metadata Schema

**Applications**
- Demand Response
- Occupant Interaction
- NILM
- Occupancy Models
- Predictive Control
- Fault Detection

**Management**
- APIs
- Data Storage
- Access Control
- Monitoring
- Search
- Privacy

**Metadata Schema**

**Buildings**
- Home
- Office
- Factory
- Apartment
- Hospital
- Shopping Mall

**Infrastructure: Sensors Equipment**
- Smoke Detector
- Thermostat
- Motion Sensor
- Fire Safety
- HVAC
- Lighting
Existing Schema Are Inadequate

BIM/IFC

Domain
Construction

Coverage
Architecture, Mechanical, Electrical Design

Data Model
EXPRESS

Limitation
Lack of sensors, equipment
## Existing Schema Are Inadequate

<table>
<thead>
<tr>
<th></th>
<th>BIM/IFC</th>
<th>SAREF, SSN</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Domain</strong></td>
<td>Construction</td>
<td>Smart Appliances, Sensor Networks</td>
</tr>
<tr>
<td><strong>Coverage</strong></td>
<td>Architecture, Mechanical, Electrical Design</td>
<td>Location, Appliance, Sensors</td>
</tr>
<tr>
<td><strong>Data Model</strong></td>
<td>EXPRESS</td>
<td>Semantic Ontology</td>
</tr>
<tr>
<td><strong>Limitation</strong></td>
<td>Lack of sensors, equipment</td>
<td>Limited vocabulary</td>
</tr>
</tbody>
</table>

*Brick, BuildSys 2016*
# Existing Schema Are Inadequate

<table>
<thead>
<tr>
<th>Domain</th>
<th>BIM/IFC</th>
<th>SAREF, SSN</th>
<th>Haystack</th>
</tr>
</thead>
<tbody>
<tr>
<td>Construction</td>
<td></td>
<td>Smart Appliances, Sensor Networks</td>
<td>Building Management</td>
</tr>
<tr>
<td>Architecture, Mechanical, Electrical Design</td>
<td></td>
<td>Location, Appliance, Sensors</td>
<td>HVAC, Lighting, Weather, Sensors, Points</td>
</tr>
<tr>
<td>Coverage</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Model</td>
<td>EXPRESS</td>
<td>Semantic Ontology</td>
<td>Tags</td>
</tr>
<tr>
<td>Data Model</td>
<td></td>
<td>Limited vocabulary</td>
<td>Inadequate tooling, limited relationships</td>
</tr>
<tr>
<td>Limitation</td>
<td>Lack of sensors, equipment</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

11/16/16  
Brick, BuildSys 2016
Existing Schema Are Inadequate

Coverage and Expressiveness in Existing Schema

- BIM/IFC: Coverage 29, Application Requirements 86
- SSN: Coverage 11, Application Requirements 41
- Haystack: Coverage 54, Application Requirements 77

Collaborative Solution to a Common Problem

- Limitations of existing schema: Feb ‘16
- Informal Workshop at Berkeley: May ‘16
- BuildSys Deadline: Jun ‘16
- Open Source Release & Demo: BuildSys ‘16
Brick: Building Metadata Schema

Applications
- Demand Response
- Occupant Interaction
- NILM
- Occupancy Models
- Predictive Control
- Fault Detection

Management
- APIs
- Data Storage
- Access Control
- Monitoring
- Search
- Privacy

Brick

Buildings
- Home
- Office
- Factory
- Apartment
- Hospital
- Shopping Mall

Infrastructure: Sensors Equipment
- Smoke Detector
- Thermostat
- Motion Sensor
- Fire Safety
- HVAC
- Lighting
Brick: Building Metadata Schema

- Completeness: Capture all sensors/subsystems
- Expressiveness: Capture relationships to run applications
- Usability: Easy to understand, easy to map buildings
Outline

- Why do we need Brick?
- Brick
  - Basics
  - Development
  - Results
- Moving forward
Outline

- Why do we need Brick?

- Brick
  - Basics
  - Development
  - Results

- Moving forward
Brick is a Graph of Building Entities
What Does Brick Contain?

Buildings
- UCSD
- IBM, Ireland
- CMU

Tags, Tagsets

Class Hierarchy

Brick

Relationships
An Example “Model” Building

- Lighting Controller
- Lighting Zone
- AHU (Return Fan, Supply Fan)
- Power Meter
- VAV (Damper)
- Thermostat
- Temperature
- CO2 Sensor
- Room 101
- Room 102
- HVAC ZONE
Brick is a Graph of Building Entities

We choose to model this graph using ontologies

- Alternatives: Verilog, IFC
Tagsets: Brick Entities
Tagsets and Tags

- **Tagsets**: Building entities described in Brick
  - “Zone Temperature Sensor”
  - “Zone CO2 Sensor”
  - “Room”
  - “Air Handler Unit” (also called “AHU”)

- **Tags**: Decomposition of tagsets
  - “Zone Temperature Sensor” -> “Zone” “Temperature” “Sensor”

- Tags facilitate annotation and keyword search
- Tagsets facilitate semantic modeling
Tagsets are Organized in a Hierarchy

**Location**
- Building
- Room
  - Conference Room
- HVAC Zone
- Lighting Zone
- ...

**Equipment**
- Fire Safety System
- HVAC
  - AHU
  - Fan
    - Supply Fan
  - Thermostat
  - ...

**Point**
- Alarm
- Command
  - Damper Command
- Sensor
  - Temperature Sensor
- Setpoint
  - Temperature Setpoint
- ...

11/16/16  Brick, BuildSys 2016
Relationships Link Tagsets

Essential to capture linkages between entities
Relationships in Example Building

- Power Meter
- HVAC Zone
- AHU
- VAV
- Damper

- **Designed to capture essential aspects in BMS**
- **Restrictions to avoid misuse:** "hasLocation" limited to Location tagsets
Relationships in Example Building

- Lighting Controller
  - controls Lighting Zone
    - isLocationOf Room 102
    - isLocationOf Room 101

- Power Meter
  - hasPoint HVAC Zone
    - feeds VAV
      - hasPart Damper

- HVAC Zone
  - feeds AHU

Legend:
- Location
- Equipment
- Point
- Relationship
Function Blocks

- Enables modularity, encapsulation, reusability
- Library of function blocks to ease development
Triples as data format

- subject predicate object
- Every entity is a URL
  brick: <http://buildsys.org/ontologies/Brick#>
- example:Berg_Hall rdf:type brick:Room
  example:Light hasLocation example:Berg_Hall

SPARQL for querying

- Pattern matches across the graph
- SELECT ?temp WHERE {
  ?temp rdf:type brick:Temperature_Sensor
}
Outline

- Why do we need Brick?
- **Brick**
  - Basics
  - Development
  - Results
- Moving forward
Empirical Methodology

Six Buildings

- UCSD
- SDU
- CMU

Six Buildings

- UC Berkeley
- UVA
- IBM, Ireland

17700 points, 630000 sqft.

Eight Representative Applications

- Model Predictive Controls
- Occupancy Modeling
- Demand Response
- Energy Apportionment
- Participatory Feedback
- Fault Detection
- Web Displays
- NILM
Iterative Development Process

Buildings
- UCSD
- IBM, Ireland
- CMU

Tags, Tagsets

Class Hierarchy

Brick

Relationships
Iterative Development Process

Map Building Metadata to Brick

UCSD
IBM, Ireland
CMU

Ground Truth Metadata

Brick

UC Berkeley
UVA

Coverage: 93.2%

SPARQL for Apps
Outline

- Why do we need Brick?
- **Brick**
  - Basics
  - Development
  - Results
- Moving forward
## Coverage Across Buildings

<table>
<thead>
<tr>
<th>Building</th>
<th>UCSD</th>
<th>CMU</th>
<th>IBM</th>
<th>UVA</th>
<th>Berkeley</th>
<th>SDU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coverage</td>
<td>96%</td>
<td>99%</td>
<td>99%</td>
<td>98.5%</td>
<td>98.7%</td>
<td>98.8%</td>
</tr>
</tbody>
</table>
### App Query Matches for Buildings

<table>
<thead>
<tr>
<th>Feature</th>
<th>UCSD</th>
<th>SDU</th>
<th>CMU</th>
<th>IBM</th>
<th>UVA</th>
<th>Berkeley</th>
</tr>
</thead>
<tbody>
<tr>
<td>Occupancy</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Energy Apportionment</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Web Displays</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>MPC</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Participatory Feedback</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Fault Detection</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>NILM</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Demand Response</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

11/16/16
Live Demo
Brick vs Haystack

Haystack
- Id: TS
- temp
- point
- sensor
- roomRef
- floorRef
- vavRef
- ahuRef
- hvac

Brick
- TS is a Temperature Sensor
- TS hasLocation Room
- TS isPointOf VAV
<table>
<thead>
<tr>
<th>Brick</th>
<th>Haystack</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Brick vs Haystack

<table>
<thead>
<tr>
<th>Reference Implementation</th>
<th>Brick</th>
<th>Haystack</th>
</tr>
</thead>
<tbody>
<tr>
<td>6 buildings, 8 applications</td>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>
Brick vs Haystack

<table>
<thead>
<tr>
<th>Reference Implementation</th>
<th>Brick</th>
<th>Haystack</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relationships</td>
<td>6 buildings, 8 applications</td>
<td>Can link entities, but does not classify relationships</td>
</tr>
<tr>
<td></td>
<td>Captures relationships within and across subsystems</td>
<td>None</td>
</tr>
</tbody>
</table>

11/16/16
Brick vs Haystack

Reference
Implementation

Brick

Haystack

Relationships

Brick: Captures relationships within and across subsystems

Haystack: Can link entities, but does not classify relationships

Querying

Brick: SPARQL queries that traverse Brick graph

Haystack: Restrictive query, cannot traverse relationships

Reference
Implementation

Brick: 6 buildings, 8 applications

Haystack: None
## Brick vs Haystack

<table>
<thead>
<tr>
<th>Reference</th>
<th>Implementation</th>
<th>Relationships</th>
<th>Querying</th>
<th>Encapsulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brick</td>
<td>6 buildings, 8 applications</td>
<td>Captures relationships within and across subsystems</td>
<td>SPARQL queries that traverse Brick graph</td>
<td>Functional blocks encapsulate complex subsystems</td>
</tr>
<tr>
<td>Haystack</td>
<td>None</td>
<td>Can link entities, but does not classify relationships</td>
<td>Restrictive query, cannot traverse relationships</td>
<td>No modularity</td>
</tr>
</tbody>
</table>
Outline

- Why do we need Brick?
- Brick
  - Basics
  - Development
  - Results
- Moving forward
Lessons Learned

- Mapping buildings to Brick
  - Semi-automated mapping: 5 papers in BuildSys 2015

- New types of building and equipment
  - Update building tagsets
  - An unknown sensor can be referred to as “Sensor”

- Separation of application from building infrastructure
  - Application specific terms should not be part of Brick
Moving Forward: Call to Action!

- Open Sourced with BSD license
  http://brickschema.org/

- Sustained improvements, extensions:
  - Comments, issue tracking, pull requests on github

- Integrate with building management systems
  - We have an initial integration with BuildingDepot

- Compatibility with IFC
  - Exploit IFC “adapters” to convert CAD specs to Brick

- Brick in Practice
  - Usability testing, performance evaluation, tool-chain support

11/16/16
Brick, BuildSys 2016
Thank You!

Visit our Demo!

Authors who are at BuildSys

Bharathan Balaji, Arka Bhattacharya, Gabriel Fierro, Jason Koh, Yuvraj Agarwal, David Culler, Mikkel Kjaergaard, Mario Berges, Kamin Whitehouse

http://brickschema.org/

Image credit: clipartpanda.com, school-pass.com, clker.com