

Lecture – 07

Intro. to Internet of Things

Dr. Ahmed Elngar

Faculty of Computers and Artificial Intelligence

Beni-Suef University

17. IoT – ThingWorx

Thingworx is a platform for the rapid development and deployment of smart, connected devices. Its set of integrated IoT development tools support connectivity, analysis, production, and other aspects of IoT development.

It offers Vuforia for implementing augmented reality development, and Kepware for industrial connectivity. KEPServerEX provides a single point for data distribution, and facilitates interoperability when partnered with a ThingWorx agent.



Components

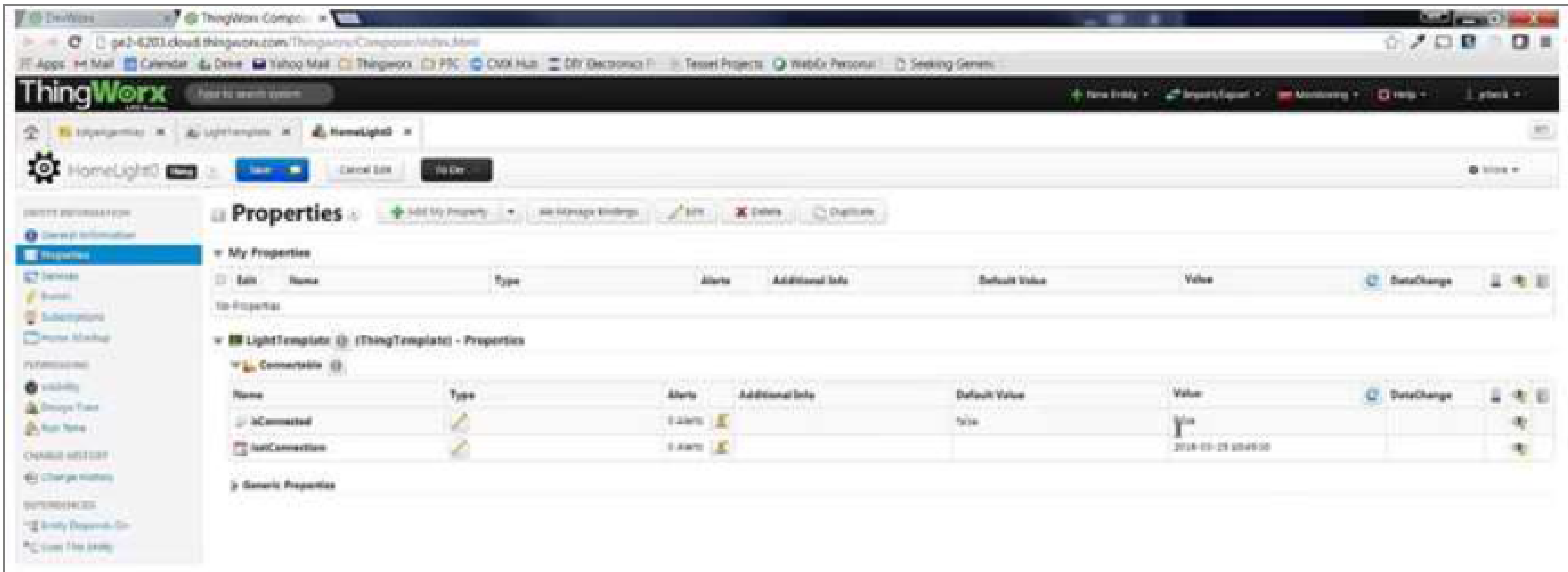
Thingworx offers several key tools for building applications. These tools include the Composer, the Mashup Builder, storage, a search engine, collaboration, and connectivity. The Composer provides a modeling environment for design testing. The Mashup Builder delivers easy dashboard building through common components (or widgets); for example, buttons, lists, wikis, gauges, and etc.

Thingworx uses a search engine known as SQUEAL, meaning Search, Query, and Analysis. Users employ SQUEAL in analyzing and filtering data, and searching records.

Interface

The ThingWorx platform uses certain terms you must familiarize yourself with. In the main screen's top menu, you search for **entities** or create them. "Entity" refers to something created in ThingWorx. You can also import/export files and perform various operations on them.

In the left menu, you find entity groups, which are used to produce models and visualize data; and manage storage, collaboration, security, and the system.



When you select the Modeling category in the menu, you begin the process by creating an entity. The entity can be any physical device or software element, and it produces an **event** on changes to its property values; for example, a sensor detects a temperature change. You can set events to trigger actions through a subscription which makes decisions based on device changes.

Data Shapes consist of one or more fields. They describe the data structure of custom events, infotables, streams, and datatables. Data shapes are considered entities.

The screenshot shows the PTC Access App Resource PLM Part Data Shape configuration interface. The browser title bar displays "PTC Access App Resource PLM Part Data Shape" with "Database" and "To Do" buttons. The main area is titled "Fields" and contains a table with columns for Name, Description, and other attributes. The table lists various fields such as UsageName, Number, Name, Description, Version, Size, Unit, Quantity, Location, LocationBody, Created, and Type. The "mySoftAttribute" field is highlighted in blue. Below the table, the configuration details for "mySoftAttribute" are shown, including a "General Field Info" section with fields for Name (mySoftAttribute) and Description, a "Base Type Info" section with a dropdown menu for Base Type (set to "String") and a field for Has Default Value, and an "Attribute" section with a checkbox for Is Primary Key.

| Name | Description | Base Type | Is Primary Key |
|------------------------|-------------|-----------|----------------|
| UsageName | | | |
| Number | | | |
| Name | | | |
| Description | | | |
| Version | | | |
| Size | | | |
| Unit | | | |
| Quantity | | | |
| Location | | | |
| LocationBody | | | |
| Created | | | |
| Type | | | |
| mySoftAttribute | | | |

mySoftAttribute

General Field Info

Name: mySoftAttribute

Description:

Base Type Info

Base Type: String

Has Default Value:

Attribute

Is Primary Key:

Thing Templates and **Thing Shapes** allow developers to avoid repeating device property definitions in large IoT systems. Developers create Thing Templates to allow new devices to inherit properties. They use Thing Shapes to define Templates, properties, or execute services.

Note a Thing only inherits properties, services, events, and other qualities from a single template, however, Things and templates can inherit properties from multiple Thing Shapes.

Development

ThingWorx actually requires very little programming. Users connect devices, establish a data source, establish device behaviors, and build an interface without any coding. It also offers scalability appropriate for both hobbyist projects and industrial applications.

18. IoT – Cisco Virtualized Packet Core

Cisco Virtualized Packet Core (VPC) is a technology providing all core services for 4G, 3G, 2G, WiFi, and small cell networks. It delivers networking functionality as virtualized services to allow greater scalability and faster deployment of new services at a reduced cost. It distributes and manages packet core functions across all resources, whether virtual or physical. Its key features include packet core service consolidation, dynamic scaling, and system agility.



Its technology supports IoT by offering network function virtualization, SDN (software-defined networking), and rapid networked system deployment. This proves critical because its virtualization and SDN support low-power, high flow networking, and the simple deployment of a wide variety of small devices. It eliminates many of the finer details of IoT systems, and conflicts, through consolidating into a single system and single technology for connecting and integrating all elements.

Use Case: Smart Transportation

Rail transportation provides a viable example of the power of VPC. The problems VPC solves relate to safety, mobility, efficiency, and service improvement:

- Rail applications use their own purpose-built networks, and suffer from interoperability issues; for example, trackside personnel cannot always communicate with local police due to different technologies.
- Determining if passengers need extra time to board remains a mostly manual task.
- Data updates, like schedules, remain manual.
- Each piece of equipment, e.g., a surveillance camera, requires its own network and power source.