Web technologies for smart manufacturing

Dr. Dave Raggett <dsr@w3.org>
W3C lead for Web of Things

Web of Things IG Open Day
Sophia Antipolis, 26 January 2016
IoT Applications

Many applications areas with a huge overall potential!

- Smart environmental control with energy savings in the home and reduced peak demand on the grid
- Wearables and augmented reality for context based notifications
- Assisted living and home-based healthcare
- Guidance to nearby parking spaces in smart cities
- Richer engagement with customers through proactive maintenance and software based product enhancements
- Smart manufacturing of bespoke products
Manufacturing: Past, Present and Future

- Disruptive changes in the past
  - From cottage industry to mass production
    - Standardization as key to cost reduction
  - Computerisation
    - Enterprise resource planning and order processing
    - Numerically controlled mills, lathes and robots
  - Globalisation
    - Low cost transport of goods and materials
    - Offshoring for cheaper labour costs
  - Cheap ubiquitous networking
    - Accelerating the pace of globalisation

- Coming next – smart manufacturing
  - Seismic shifts as companies embrace the IoT
    - Michael Porter – How Smart Connected Products are Transforming Competition
The Winds of Change

Drivers of Industry 4.0

Flexible manufacturing operations help fulfill customer wishes even better.

- Shorter delivery times
- Volatile markets
- 24/7 service
- Shorter product life cycles
- More individualized customer wishes

Source: Bosch
Smart Manufacturing

• Shift from mass production to bespoke production
  – Emphasis on tailoring finished products to match each customer's unique needs
    • Focus on design skills
    • Mass production for “standard” parts
  – Reduced time from design to delivery
    • Re-shoring of production
  – Flexible production systems that can be rapidly reconfigured to meet changing needs
  – Open markets of services
• Smarter systems
  – Importance of models and metadata
    • Production planning
    • Monitoring and optimisation
    • Cost reduction
    • Vertical integration across different levels of abstraction
    • Horizontal integration across different functional areas
  – Integration along the value and supply chains

Tesla’s new production line
Bespoke Products

- Divide manufacturing plant into cells for efficiency
- Specialised Cells for particular operations
  - e.g. lathing, milling, drilling, laser cutting, 3D printing, joining and assembling, finishing, painting, testing and inspection, packing, . . .
- Flexible Cells with Robots for general operations
  - With interchangeable tools for different tasks
- People working safely in collaboration with machines
  - Robots doing strenuous repetitive work leaving people to do the lighter, more intricate tasks
- A static production line could be too restrictive
  - Won't work well when each product requires a different sequence of operations
- Instead, use transport robots to move materials between cells
  - Semiconductor fabrication plants
    - Overhead tracks
  - Fully automated warehouse
    - Amazon as a modern example
  - Allocating cells to operations to maximise utilisation
    - Avoiding cells idling waiting for other cells
    - Allowing for maintenance of production cells
  - How to optimise the movement of materials between cells?
Production Layout

• Choices depend on what your making
  – Fixed Position plant layout
    • Product stays put and resources move to it
      – e.g. building a ship in a dry-dock
  – Product oriented plant layout
    • Machinery and materials are placed following the product path through the factory
      – e.g. building a car on an assembly line
  – Process oriented plant layout
    • Machinery placed according to what they do and materials go to them
      – e.g. brewing beer on a commercial scale
  – Combined layout
Choose Your Robot

Robots for cutting, welding, painting, moving, helping, ...
Digital Automation Technologies

A confusing landscape ...

- **Fieldbus**
  - A suite of protocols for connecting programmable logic controllers to sensors and actuators at the field level in a hierarchy of control systems
  - Wide variety of competing fieldbus standards, e.g.
    - **HART**
      - Highway Addressable Remote Transducer Protocol – an early implementation of Fieldbus – it remains a very popular digital automation protocol
    - **PROFIBUS**
      - PROFIBUS SP for operating sensors and actuators from a centralised controller
      - PROFIBUS PA for monitoring equipment in process automation
    - **Modbus**
      - Commonly used suite of protocols originally designed for programmable logic controllers, it is master/slave based, and requires the master to poll the field devices
  - **CAN**
    - Controller Area Network for information exchange between microcontrollers
  - **DeviceNet**
    - Network technology for interconnecting control devices, using CAN as backbone
  - **EtherCAT**
    - Ethernet based fieldbus designed for low latency and jitter

- **MTConnect**
  - Protocol for reading data from control devices, e.g. numerically controlled machine tools

- **OPC, OPC-UA**
  - Open Platform Communications (OPC) was originally designed for communication between Window's based applications and process control hardware, OPC services use Microsoft's Component Object Model (COM) for client-server communication
  - OPC-UA covers an information model with nodes, events and methods, as well as a choice between a binary protocol and an XML based protocol (SOAP over HTTP)

- **ANSI/ISA-95**
  - Interface between enterprise and control systems
  - Consistent terminology between suppliers and manufacturers
  - ISA-95 models for context, scheduling, object relationships and attributes, operation activities
Evolution not Revolution

- Companies have a big investment in existing technologies
  - The equipment and the expertise in operating them
- Gradual introduction of model based techniques
  - Models of production equipment and materials
  - Support for production planning
  - Integration with live data feeds
- Role of Web Technologies for bridging platforms
  - Uniform framework for semantics, data & metadata
  - Abstraction layer to simplify application logic
  - Bindings to existing digital automation technologies
Traditional Hierarchy for Industrial Automation

From: S. Djiev, *Industrial Networks for Communication and Control*
• German initiative for smart manufacturing
  - Platform40.de
Industrie 4.0

RAMI

- Industrie 4.0 – High Level Architecture
Industrie 4.0 Components

- Fundamental building block for Industrie 4.0
- Unified model for description of assets
  - From sensor or actuator to the whole plant
- Consists of asset enriched by an administrative shell
  - Virtual representation of the real asset
  - Holds data for asset's status and data generated during asset's lifecycle

I4.0 Component = “Thing” in Web of Things
Web of Things as Enabler for Vertical & Horizontal Integration

Integration along the supply chain

Integration along the value chain*

Web of Things
(I4.0 “components”)

High levels of abstraction

Low levels of abstraction

Business Level

Field Level
(I4.0 “assets”)

- Distributed services
- Platform of platforms
- Uniform addressing
- Data and metadata

*Value Chain – The process or activities by which a company adds value to an article, including design, production, marketing, and the provision of after sales service
Lowering the Barrier for Integration

**Apps for Smart Manufacturing**

- Large companies want their suppliers to integrate with their software systems for greater efficiencies along the value chain
- SMEs find this challenging – the cost of developing the corresponding software is a significant barrier
- The Web of Things can simplify this through an abstraction layer above existing systems
- Open markets of services as a way of commoditising the building blocks
- Replacing Monolithic software with Manufacturing Apps
Open Standards for Open Markets of Things

- Connecting suppliers and consumers
  - Integration along the supply and value chains

- Software and hardware
  - As individual components
  - As assembled systems

- Different business models
  - Direct purchase, subscription, pay as you go, ...

- Hosted stores vs fully distributed marketplaces
  - Open standards to counter fragmentation

- Marketplace features
  - Discovery, reviews, recommendations, ranking/reputation
  - Dynamic composition to match given requirements
  - Automated negotiation of contracts to save time & money

- Lifecycle support
  - Developing, testing, publishing, vetting, updates, obsolescence
Automated Negotiation

• Negotiation over contracts
  – B2C: typically a fixed contract
    • Exception with auctions like eBay
  – B2B: negotiation over price, terms & conditions
  – Aim to reach mutually satisfactory agreement
  – May involve protracted discussions

• Automated negotiation can save time and money
  – No need to involve teams of people on each side
  – Applicable to markets of dynamic machine to machine services
  – Many forms possible, e.g.
    • Series of rounds between suppliers and consumers
    • Auctions, e.g. where suppliers tender for consumer's request
  – Jennings et al. (2001) frame automated negotiation in terms of
    • Negotiation protocols that govern the interaction between the negotiating parties
    • Negotiation objects defining the range of issues over which agreement must be reached
    • Agents’ decision making model
Next Steps?

- Potential topics of interest for common work
  - Uniform framework for data and metadata
    - RDF and its relationship to IEC 61360 (PLIB)
    - Situational awareness, planning, optimisation, ...
  - Small set of use cases and associated requirements as basis for analysis
    - Security and resilience
    - Communication patterns and requirements
      - e.g. real-time requirements at different levels of abstraction
    - Markets of services as a way to reduce costs and increase flexibility
      - Automated negotiation of contracts

- Initiate a group to focus on smart manufacturing
  - Cooperation with I4.0 and Industrial Internet Consortium
Web of Things
Web of Things

- The Web of Things is designed to appeal to developers by reducing the cost and complexity for creating services
  - Clean separation of abstraction layers
- A uniform approach to services that scales across devices, platforms and domains
- This will pave the way to an open market for services on the scale of the Web
  - Increased market size for solutions
- We're reaching out to developers to help W3C to address the challenges and lead the Web of Things to its full potential!
Things

- Applications act on software objects that stand for “things”
  - Software objects on behalf of local “things”
    - Sensors and actuators connected to this device
  - Software objects as proxies for remote “things”
    - Enables distributed control
  - Servers manage communication along proxy chains
    - Using the metadata associated with a “thing”

- Every “thing” has a URI for its name
  - As per Resource Description Framework
  - Used for the thing's data model, data, semantics & other metadata
  - Ontologies that describe “things”
Why is Semantics Important?

- What is the relevance to digital automation?
  - Shared standardized vocabularies for entities and their relationships
  - Describing the software objects that stand for "things"
  - Verifying that a data source and sink are compatible and have the same semantics
  - When searching for services with a given semantics
  - To facilitate the design of service compositions
  - Optimal planning for flexible production of bespoke products
  - To enable simulation prior to deploying changes to cyber-physical systems
  - To enable fault diagnosis based upon causal models

With thanks to Fabien Gandon

W3C Semantic Web Standards Stack

RDF triples: <Subject, Predicate, Object> globally unique names using URLs that act as links
Distributed Web of Things

- Thing descriptions can be used to create proxies for a thing, allowing scripts to interact with a local proxy standing for a remote entity
  - Web page scripts in browser can create proxies for things on servers
Web Servers at Many Scales

Web of Things servers can be realised at many scales from microcontrollers to clouds

- **Home Hub**: home/office server for access to smart home and wearables, running behind firewall
- **Micro-controller**: resource constrained, IoT devices or gateways, CoAP, running behind firewall
- **Smart Phone**: personal server for access to smart home and wearables
- **Cloud-Based**: highly scalable server for many users, devices and working with big data

Servers are free to choose which control languages they support
Precompile service behaviour for constrained devices
Web of Things

- Making life easier for application developers by providing a **simple scripting model**
  - **Things** standing for physical and abstract entities
  - **Things** as software objects with **properties, actions** and **events**
  - **Things** with **rich descriptions** formalised in RDF

- Server creates software object based upon thing's description
  - Server handles communication between things

- Applications scripts **decoupled** from underlying protocols which can be selected according to need
  - **Servers can further choose** which **communication patterns** to use, e.g. push, pull, pub-sub and peer to peer as appropriate
  - Potential for **multiplexing data** from multiple sensors
  - Potential for **buffering sensor data** (optimise battery, network)
  - Dealing with **battery operated devices that sleep a lot**
  - Dealing with **real-time requirements**

- Reliant upon **shared semantics and rich metadata**
  - Using W3C's Resource Description Framework
# Abstraction Layers

<table>
<thead>
<tr>
<th>Application</th>
<th>Scripts that define thing behaviour in terms of their properties, actions and events, using APIs for control of sensor and actuator hardware</th>
</tr>
</thead>
</table>
| Things      | Software objects that hold their state  
Abstract thing to thing messages  
Semantics and Metadata, Data models and Data |
| (I4.0 Components) |                                                                                                                                   |
| Transfer    | Bindings of abstract messages to mechanisms provided by each protocol, including choice of communication pattern, e.g. pull, push, pub-sub, peer to peer, ... |
| Transport   | REST based protocols, e.g. HTTP, CoAP  
Pub-Sub protocols, e.g. MQTT, XMPP  
Others, including non IP transports, e.g. Bluetooth |
| Network     | Underlying communication technology with support for exchange of simple messages (packets) |

Standards are needed at each layer
Horizontal and Vertical Metadata

Smart Homes
Smart Lifecare
Smart Cities
Smart Industry

Core Metadata used across application domains

Industry specific groups are in best position to define metadata for each vertical
W3C view of Horizontal Metadata

Core metadata applicable across application domains

- **Thing descriptions**
  - Links to thing semantics
  - Data models & relationships between things
  - Dependencies and version management
  - Discovery and provisioning
  - Bindings to APIs and protocols

- **Security related metadata**
  - Security practices
  - Mutual authentication
  - Access control
  - Terms & conditions
    - Relationship to "Liability"
  - Payments
  - Trust and Identity Verification
  - Privacy and Provenance
  - Safety, Compliance and Resilience

- **Communication related metadata**
  - Protocols and ports
  - Data formats & encodings
  - Multiplexing and buffering of data
  - Efficient use of protocols
  - Devices which sleep most of the time
To realise the full potential for the Web of Things, we will need to reach out to many communities.
Discussion?
Further Details
Benefits of the Web of Things

- Reduced costs and increased market size through open web technology standards for an open market of services
  - Counter fragmentation and realize the massive potential!
- W3C is unique in tackling the IoT from the perspective of app developers
  - Freeing app developers from complexities of protocols, message formats and communication patterns
  - These can be chosen to match specific needs of each context
- Our approach is a platform of platforms
  - Making it easy to build services spanning devices from microcontrollers to cloud based server farms
  - Services spanning platforms from different vendors and different technology standards
  - Services that bridge domains for exciting new applications
  - Simple approach to scripting, together with rich models for semantics and metadata, based upon W3C's proven strengths with the Resource Description Framework and a suite of associated standards, e.g. OWL, RDF-S, SPARQL, RIF, ...
Challenges

- **Open Standards** for vocabularies for semantics, metadata and data models
  - Key to enabling web scale markets for services

- **Discovery** of services
  - The benefits of a *lingua franca*, and its limitations

- **Composition** of services
  - From different vendors for an open market of services

- **Monetization** of services
  - Support for a wide variety of models

- **Security**, privacy, safety, compliance, trust, resilience

- **Scaling** on multiple dimensions
  - Scaling across **devices** from microcontrollers to massive cloud based server farms
  - Scaling across **platform and services** from different vendors and built upon different standards
  - Scaling across application **domains**

- The inevitability of evolutionary change in complex ecosystems
  - Weakly coupled communities will evolve independently
  - How to support “trade” across these communities
Web of Things Topologies

- The Web of Things lends itself to different topologies
  - **Peer to Peer**
    - Devices talk directly to one another
    - Each device can host a mix of things and proxies for things on other devices
  - **Peer to Peer via Cloud**
    - Using a message routing network
    - Using WebRTC data channel
  - **Star** – Hub as controller for cluster of devices
    - The hub has proxies for the things on each of the devices
  - **Device to Cloud**
    - Device registers things on Cloud-based server
  - **Star to Cloud**
    - Hub acts as gateway between devices and the cloud
Intelligence in Depth

• Abstraction layers for **sensing**
  – Progressive stages of interpretation
    • Combining sensor data with other sources of information
    • Inferred events
    • Machine learning
  – Monitoring to check all is well
  – Reducing the burden on cloud based systems

• Abstraction layers for **actuation**
  – Progressively map high level intent to low level actuation
  – Synchronisation across clusters of devices

• Abstraction layers for **control**
  – Control links sensing to actuation
  – Implementing control at multiple levels of abstraction
Security & Resilience

- Safety and compliance
  - Ensuring that cyber-physical systems are safe and comply with all applicable legislation and guidelines

- Secure by design
  - Strong authentication
    - New W3C WG’s on web authentication & hardware based web security
  - Encrypted communications
  - Tamper proof storage of keys and credentials
  - Best practices for provisioning and security updates

- Resilience is about coping with faults, demand spikes and cyber-attacks
  - How to predictably and gracefully adapt to threats
  - System level response
  - Need for monitoring to detect problems
    - Trip-wires
    - Abnormal behaviours
  - Defence in depth with security zones
Provisioning and Updates

- The challenge involved in setting up devices and services
  - Establishing the security context
    - Creating a identity for the device, generating and registering its public key(s)
    - Setting up the trust relationships, e.g. for software updates
    - Threat models for provisioning

- Secure software updates
  - Essential for strong security
    - To fix security bugs, and to update security settings in line with best practices
    - To address evolving software interfaces exposed by services
  - Pushed or Pulled
  - Authenticating updates
    - Using previously provisioned public keys for trusted sources
  - Streaming updates (too large to fit in RAM)
  - Recovering from errors
    - Temporary network problems resulting in incomplete updates of Flash memory
    - Boot loaders need to test for successful updates and enter a safe update mode if previous update failed to complete
    - Implications for the security context