SWE and IoT
SWE-IoT Workshop

80th OGC Technical Committee
Austin, Texas (USA)
Dr. Mike Botts
March 21, 2012
Purpose of presentation

• Explore characteristics of Internet of Things (IoT)

• Provide overview of how OGC Sensor Web Enablement (SWE) standards could play a role in IoT

• Point to SWE 2.0 technology/design concepts with relevance to IoT

• Discussion of directions for SWE that might take to improve its ability to meet needs of IoT
Characteristics of IoT (as misunderstood by me)

- “things” are uniquely identifiable on the web
- “things” are distributed, existing in the cloud
- “things” are loosely coupled (or uncoupled) but can participate with other things in global decision processes
- “things” don’t necessarily participate actively, but are passive and “participate” by being used by agents
- “things” can cause events that initiate autonomously-driven decision processes
- “things” support sensing and action; “things” can be sensors and actuators
- “things” are often associated with small footprints, embedded software, low-energy requirements, and mesh networks
A few thoughts (according to my misunderstanding)

• “identifiable” sometimes implies RFID, bar codes, or QR codes, but not always

• “things” should not just be “identifiable” but “qualifiable”

• information is “sensed”, but not always by a physical sensor

• information is “sensed” by processes, both physical and non-physical (process concept of SensorML)
Characteristics of SWE (of relevance to IoT)

• SWE web service approach provides a decoupled approach where objects can jointly participate in a wide range of decision workflows

• SWE supports both detectors and actuators
  – SensorML – supports both
  – SOS – observations (sensors, models)
  – SPS – tasking (sensors, models, actuators)

• SWE services are self describing

• SensorML makes things qualifiable and not just identifiable

• SWE 2.0 services and encoding can support more lightweight implementations
SWE 2.0 Enhancements -1- (of relevance to IoT)

• All services and encodings
  – Conformance classes (use what you need)
  – Better use of SWE Common Data and its highly efficient data encodings and streaming

• SensorML
  – Supports sensors, actuators, computational processes
  – Required uniqueID for all processes
  – Inheritance
    • much smaller, cleaner instance descriptions
    • still supports ability to drill down to greater information

• Data interface
  • Supports efficient real-time streaming bypassing SOS
  • Supports plug-n-play sensors (IP, RS232, USB, PUCK, etc.)
  • Supports streaming from network of sensors and multiplex messaging
SWE 2.0 Enhancements -2- (of relevance to IoT)

• SOS
  – Conformance classes allow for smaller, simpler services
  – Better use of SWE Common
  – getResult/insertResult – supports efficient retrieval of values only, without bulky XML
  – Can support real-time streaming

• SPS
  – Conformance classes allow for smaller, simpler services
  – Better use of SWE Common
  – supports efficient sending of tasking values only, without bulky XML
  – Can support real-time streaming of task commands
Directions

• Education – need better guidance on efficient use of SWE for IoT community

• MUST improve discovery
  – Registries are one solution (sensors and services)
  – Peer-to-peer is perhaps more promising

• Provide lightweight profiles for services and encodings
  – e.g. in-situ, static sensor profile

• Introduce more software (services and web clients)
  – Easily configurable
  – Easily deployed
Conclusions

- SWE offers many capabilities that can and should support the emergence of IoT
- Several features in SWE 2.0 improve support for IoT
- SWE DWG should consider how to better support IoT in future