Methodology for Development and Objective Comparison of Architectures for Networked RFID

Béla Pátkai, Damith Ranasinghe,
Mark Harrison, Duncan McFarlane

Distributed Information and Automation Lab / AutoID Lab
Institute for Manufacturing, University of Cambridge

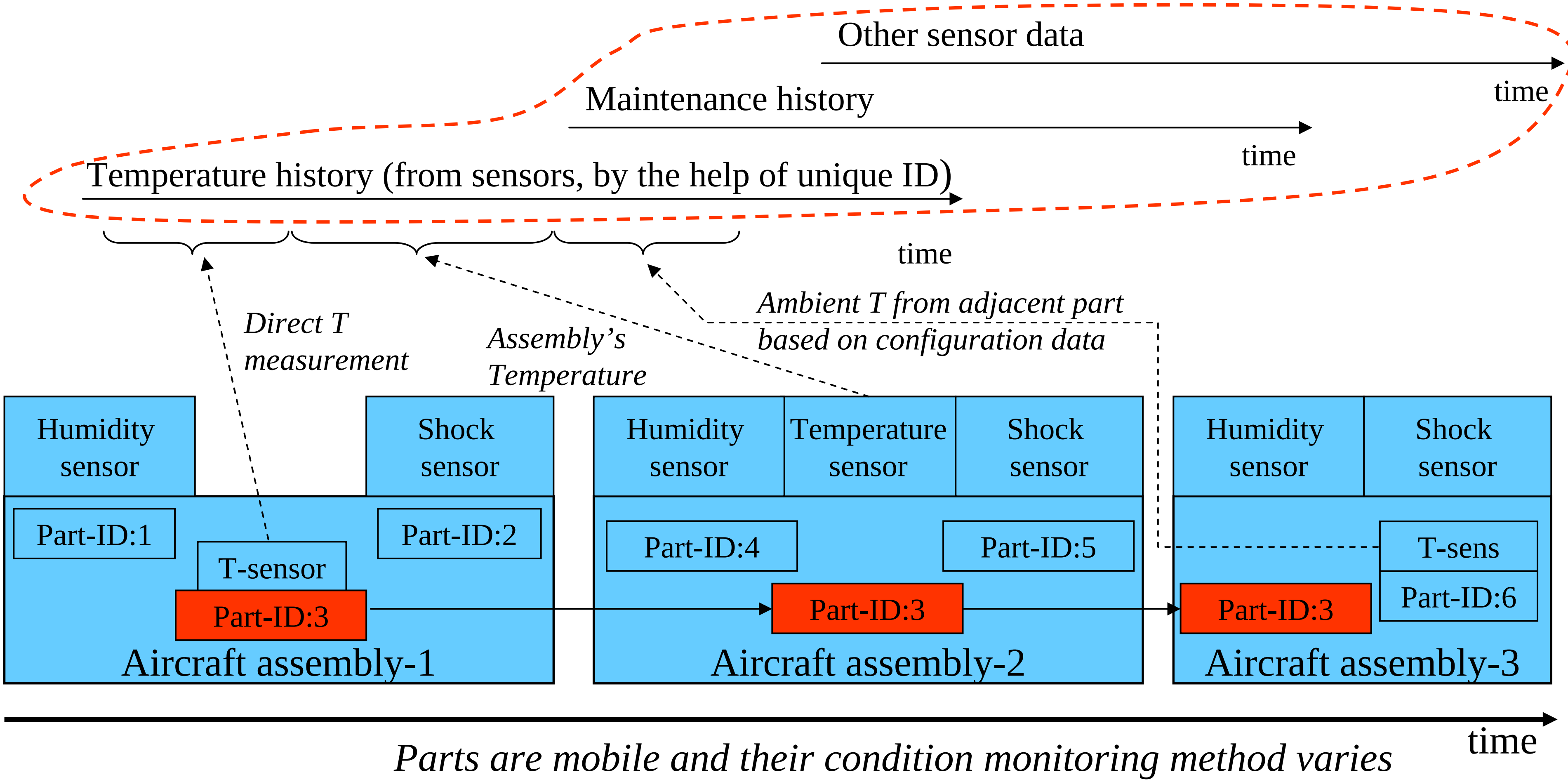
Problem Definition

- *Networked RFID is going to be implemented in a heterogeneous environment*
  - Not following a single standard
  - Different vendors
- Heterogeneous systems will have to interface
- Ad-hoc solutions
- Services easy to implement - difficult to deploy & maintain
  - Solution performance is not foreseeable
    - Computational burden
      - E.g. sensor data processing
    - Network load
    - Storage requirements
  - Indirect security issues: *accessing data allows data mining*
- Systematic work requires a methodology
- Efficient communication formalism is necessary for system/services development
- Automation of interfacing (or part of process) very desirable (M2M)

Relevance to Logistics and Motivation

- Logistic networks
  - Naturally cross-organizational
  - Very dependent on each other
  - Use a variety of IT infrastructure
  - In a different point of IT development
  - Have different IT outsourcing policy
  - Their business model may benefit a lot from improved information flow
- Logistic networks require addressing issues in the previous “Problem Definition”

Aerospace Example



Approach

- Networked RFID systems consist of a fixed set of components
- ...hence using an ontology is straightforward, except
  - The relationship between components and specific properties can vary a lot
- General design principles are available
  - Success stories (most well-known)
    - Internet
    - UNIX
  - "Design for interfaces" principle
  - IT system design methods and tools

Ontology Languages

- UML is possible
  - Limitations
  - ATL UML2OWL addresses automatic conversion
- RDF(S) (Resource Description Framework (Schema)) is used in this paper
  - Developed by the World Wide Web Consortium
  - The closest to the “Internet of Things” concept
  - Relatively well-known
  - XML-based → machine readable
- There are many other options including
  - OWL – based on RDF(S)/XML, extra functionality for reasoning
  - OKBC (Open Knowledge Base Connectivity)
  - SHOE (Simple HTML Ontology Extensions)
  - DOGMA (Developing Ontology-Grounded Methods and Applications)

The RDF Model

- Triples: <Subject> <Predicate> <Object>
- www.example.org <has_creator> <John Smith>
- More specifically
  - <http://www.example.org/index.html> <http://purl.org/dc/elements/1.1/creator> <http://www.example.org/staffid/85740>
- Multiple such statements describe resources
- Represented by
  - directed, labelled graphs
  - XML documents, schema, URIs
- XML qualified name shorthand notation = QName
  - prefix ex:, namespace URI: http://www.example.org/
  - prefix exstaff:, namespace URI: http://www.example.org/staffid/
  - prefix dc:, namespace URI: http://purl.org/dc/elements/1.1/
- Hence the QName expression is
  - *ex:index.html dc:creator exstaff:85740*

RDF Example – triples and labelled graph[1]

1.
   1. URI of EM
   2. personalTitle
   3. Dr.
2.
   1. URI of EM
   2. mailbox
   3. em@w3.org
3.
   1. URI of EM
   2. fullName
   3. Eric Miller
4.
   1. URI of EM
   2. Type
   3. Person

IfM

UNIVERSITY OF CAMBRIDGE

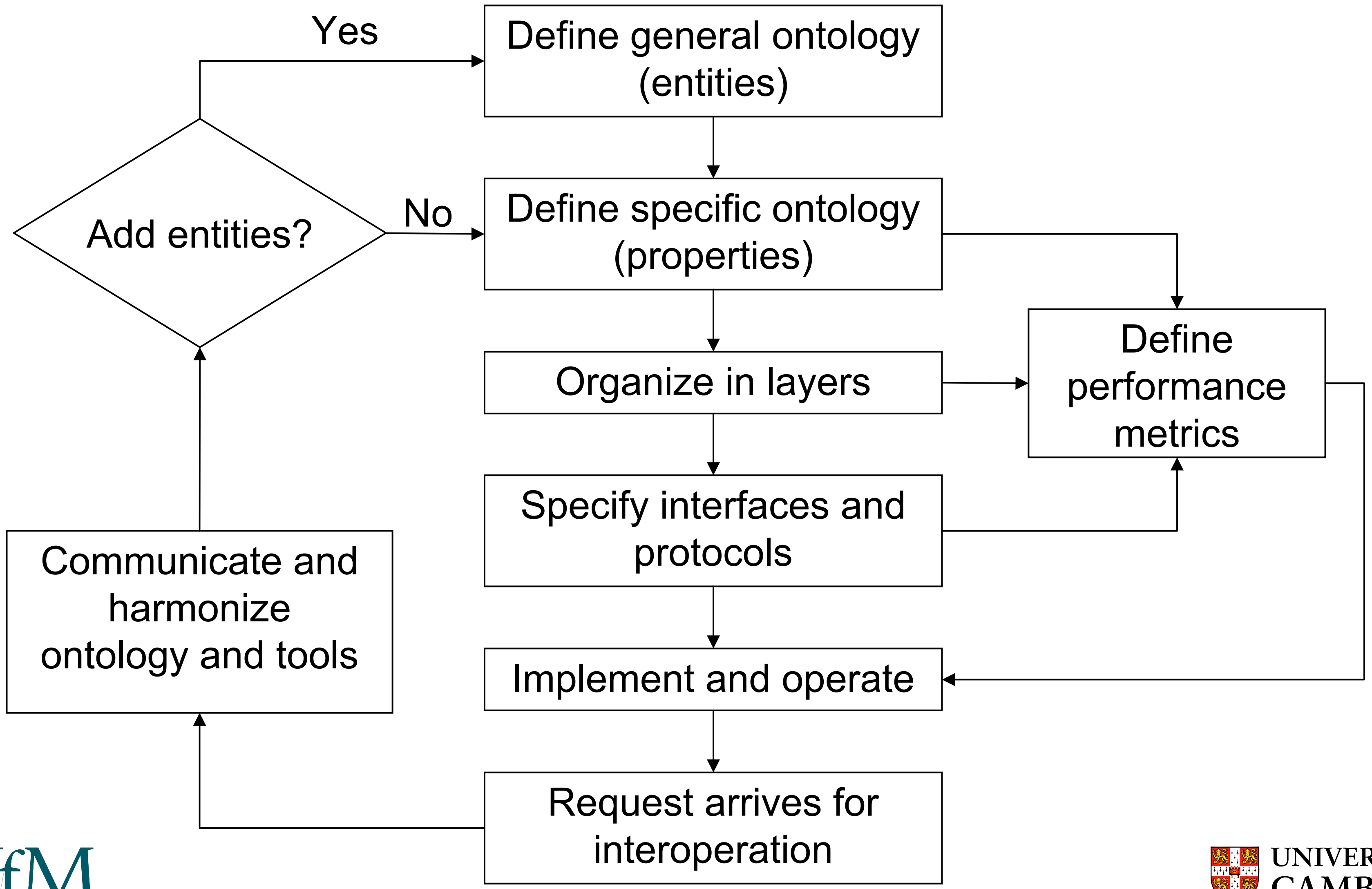
RDF/XML Syntax

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
             xmlns:contact="http://www.w3.org/2000/10/swap/pim/contact#">

  <contact:Person rdf:about="http://www.w3.org/People/EM/contact#me">
    <contact:fullName>Eric Miller</contact:fullName>
    <contact:mailbox rdf:resource="mailto:em@w3.org"/>
    <contact:personalTitle>Dr.</contact:personalTitle>
  </contact:Person>

</rdf:RDF>
```

Ontology-centred system development



IfM

UNIVERSITY OF CAMBRIDGE

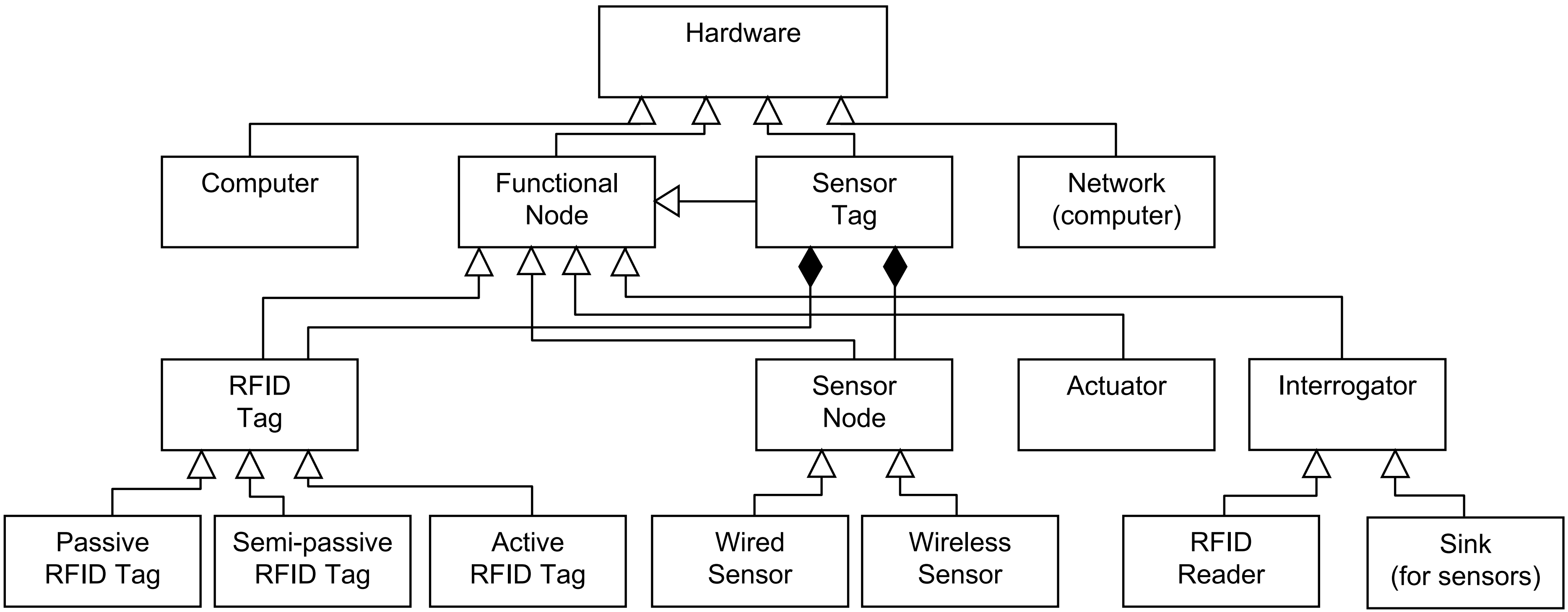
Basic Ontology
(component list with no detailed relationships)

- Hardware
  - Functional node
  - Sensor tag
  - Computer
  - Network

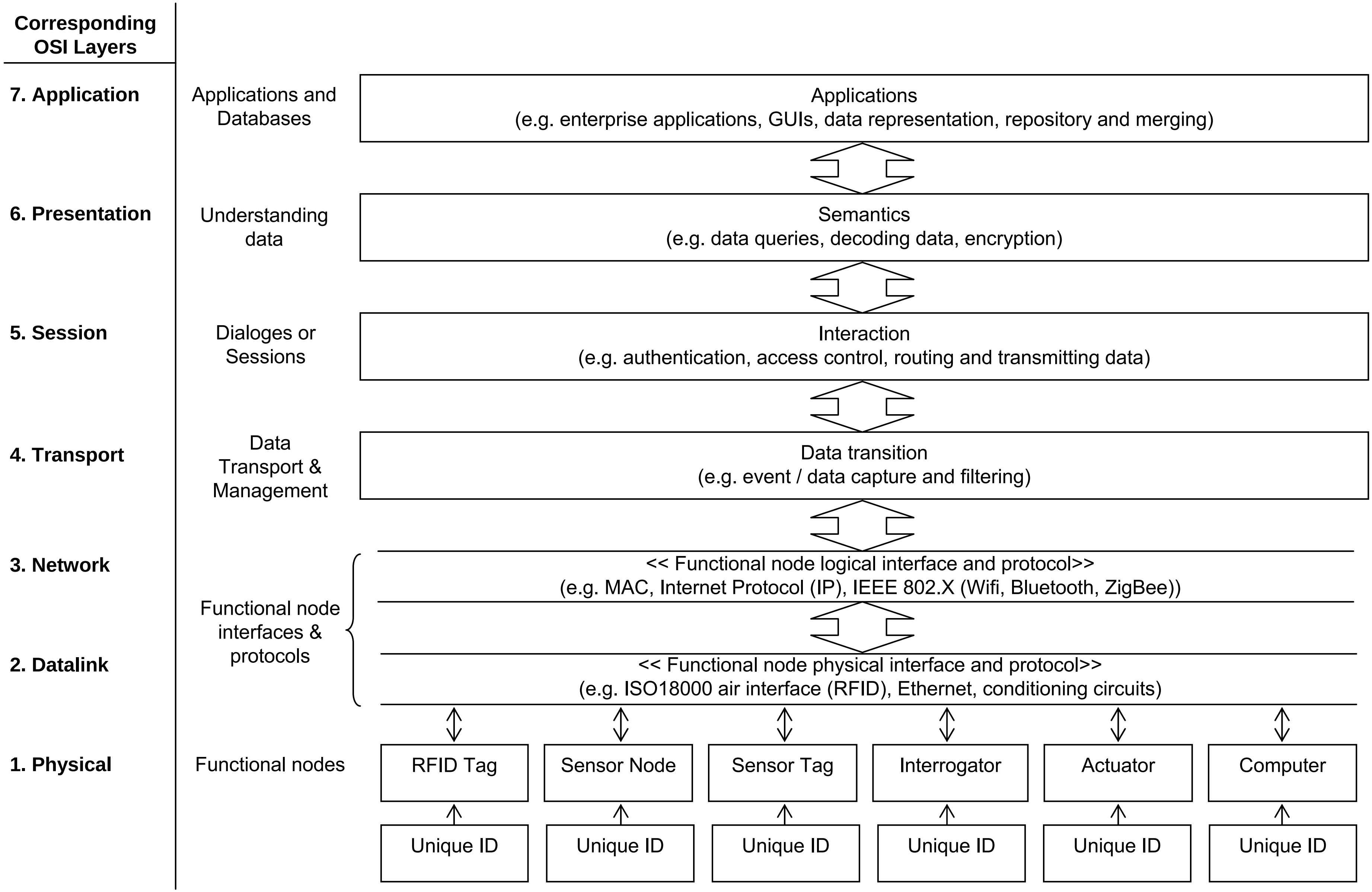
- Software
  - Data
  - Logic

- Functional node
  - RFID tag
    - Passive
    - Semi-passive
    - Active
  - Sensor node
    - Wired
    - Wireless
  - Interrogator
    - RFID reader
    - Sensor sink
  - Actuator

Hardware Ontology in UML

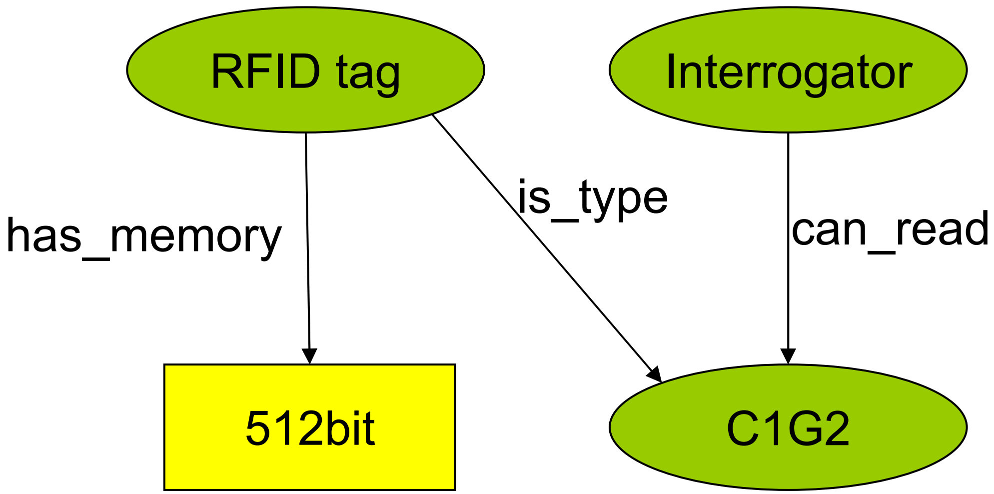


Correspondence of OSI and Networked RFID



Example: RDF triples and labelled graph

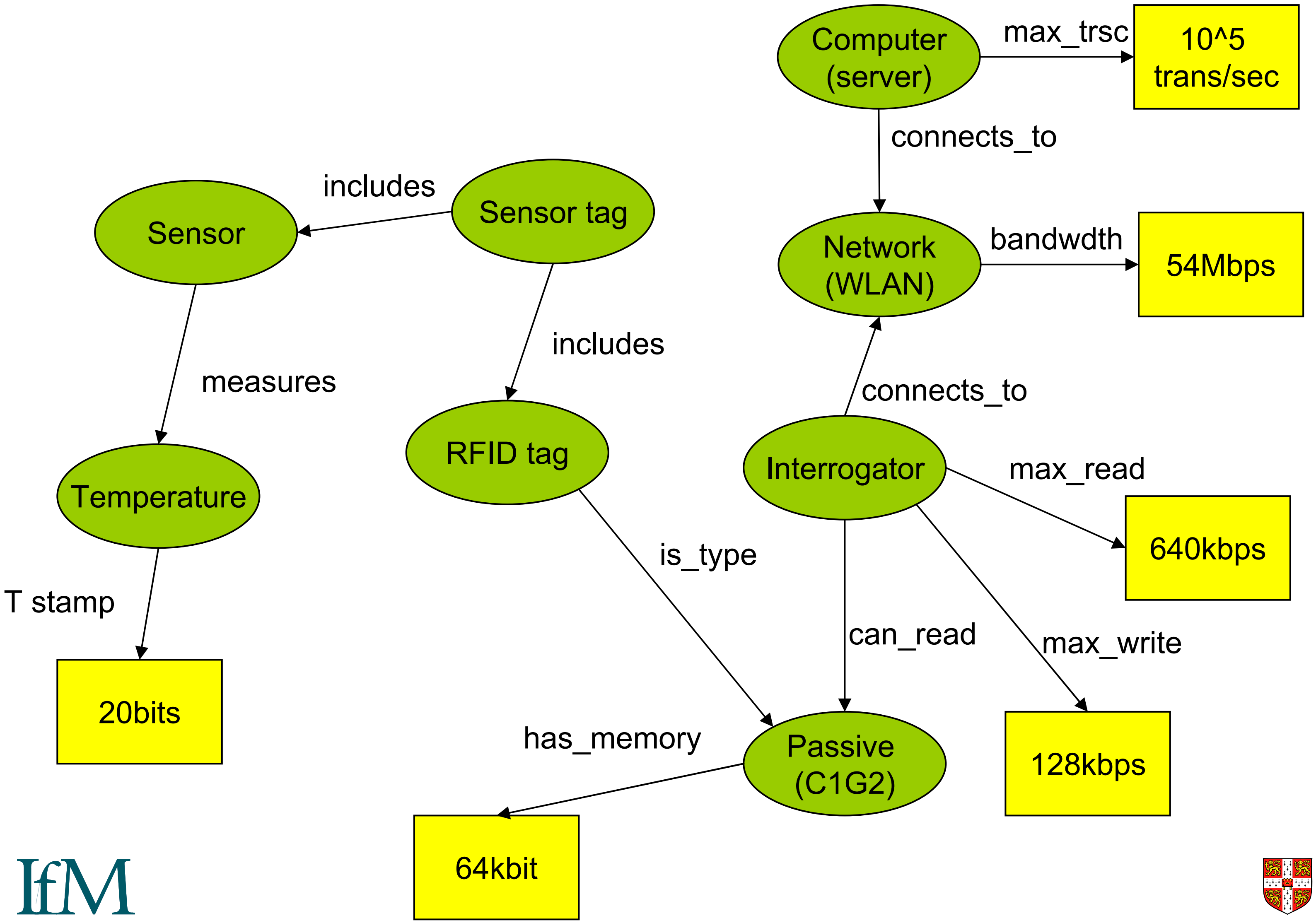
1.
    1. RFID tag (has URI)
    2. Is_type
    3. C1G2
2.
    1. RFID tag
    2. Memory_size
    3. 512 bit
3.
    1. Interrogator
    2. Is_type
    3. C1G2
4.
    1. Interrogator
    2. Can_read
    3. C1G2



Application Example

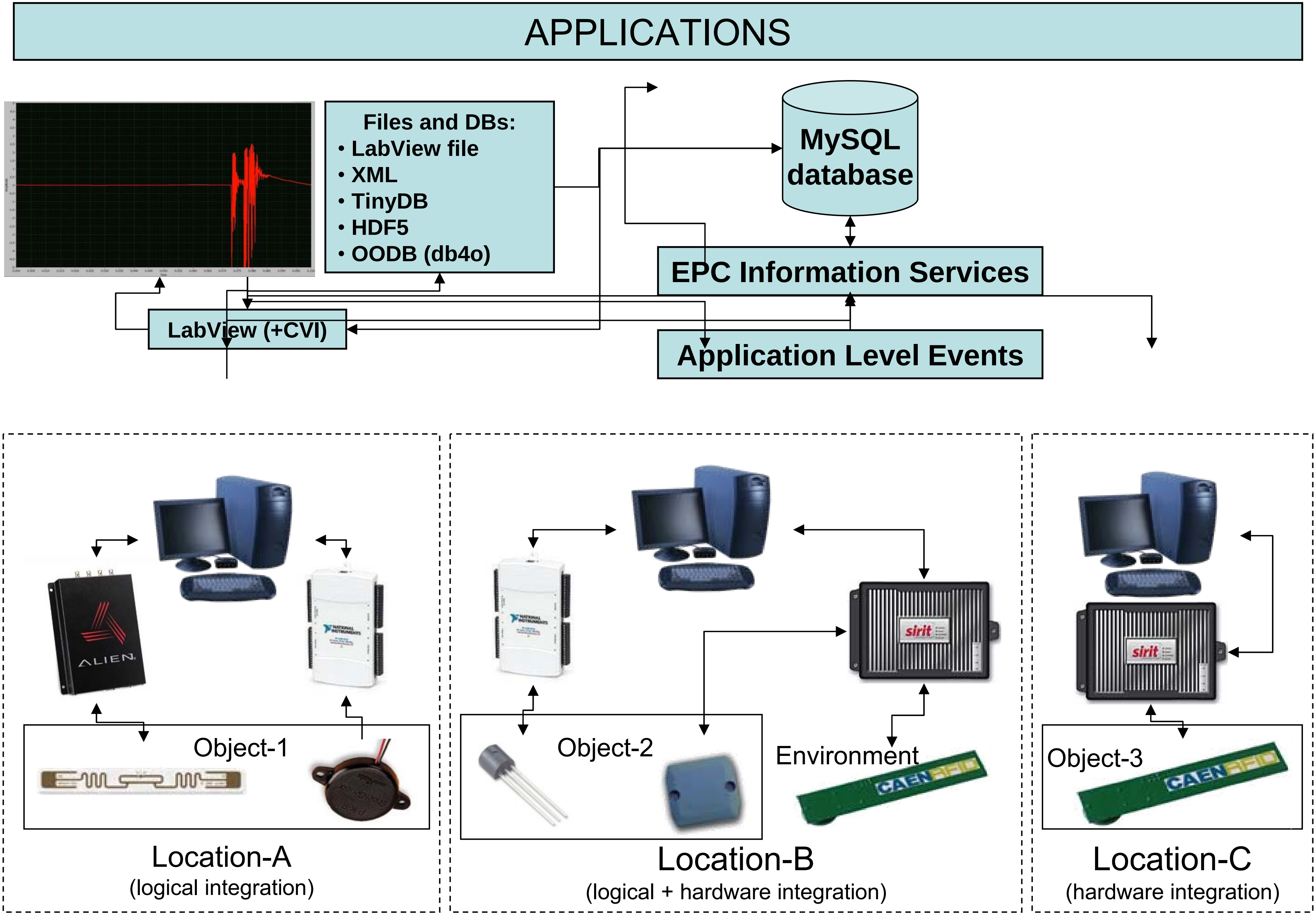
- Firms with a logistic links
  - Use bar codes and ASN on items
  - Want to introduce RFID on items for track and trace
  - Want to add condition monitoring features to their system
- The ”systematic” process (ref.slide_10)
  - They both develop and ontology model
  - Harmonize it to have a mutual understanding
  - Derive conclusions from the model
    - Performance of the integrated system
    - Bottlenecks
    - Resource usage and cost
  - Iterative design steps

Application Example – Ontology



Application Example – Performance

- Visible from previous figure
  - In case a shipment arrives with 1000 sensor tags every hour, can we share full temperature history with shipping partners?
  - Do we need more readers, servers, bandwidth?
  - Should we delgate sensor data filtering downwards?
- From a more extensive model
  - In case we have full service history of aircraft parts and exchange 500 parts per day with partners, can we sell them information and allow them to query our SQL database?
  - Should we sell raw data or invest in processing/mining?



Integration of RFID and Sensors – Lab Infrastructure

Summary and Conclusions

- Networked RFID and related infrastructure to emerge in a heterogeneous environment
- To avoid chaos methodology necessary
- Using ontologies has a number of advantages
  - Efficient communication
  - Development
  - Simulation
  - M2M
  - Agent representation
- An ontology-centred methodology was proposed