Ontology based Context Modeling

Hitesh Maidasani

CMSC818G Spring 2014

An ontology for context-aware pervasive computing environments

Harry Chen, Tim Finin, and Anupam Joshi

What is an ontology?

- a formal representation of knowledge as a set of concepts and relationships between those concepts within a domain

Why use ontologies?

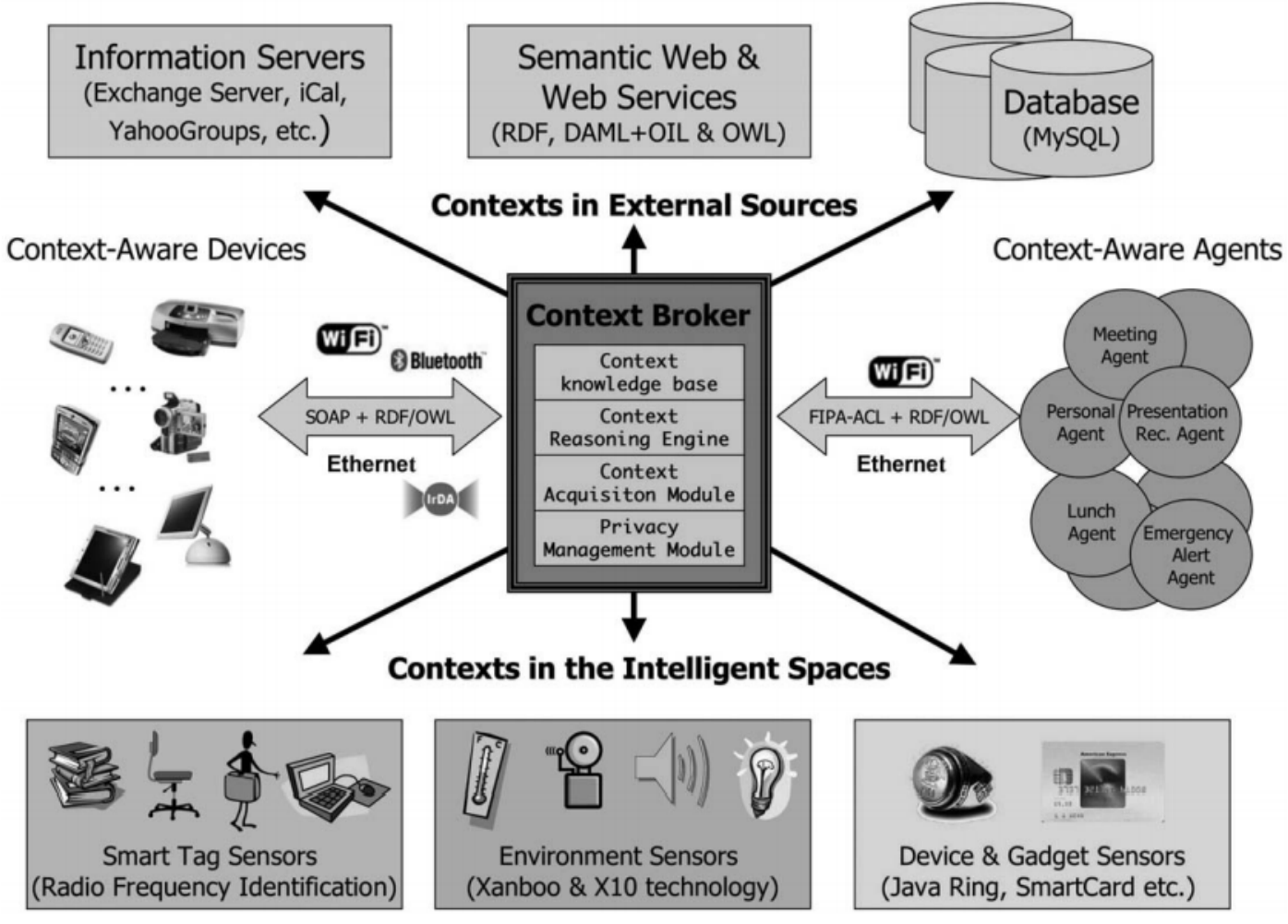
- a common ontology enables knowledge sharing in open and dynamic systems
- better reasoning of context
- allow devices and agents to work together
- **Define Context FORMALLY**

CoBrA

- Context Broker Architecture
- a collection of ontologies called COBRA-ONT for modelling the context in an intelligent meeting-room environment

Context broker architecture

- central intelligent agent called the context broker
- maintain a shared model of context
- community of agents and devices in the space
- protect the privacy of users


Information Servers
(Exchange Server, iCal, YahooGroups, etc.)
Semantic Web & Web Services
(RDF, DAML+OIL & OWL)
Database
(MySQL)
Contexts in External Sources
Context-Aware Devices
Context-Aware Agents
Context Broker
Context knowledge base
Context Reasoning Engine
Context Acquisiton Module
Privacy Management Module
WiFi
Bluetooth
SOAP + RDF/OWL
Ethernet
IrDA
WiFi
FIPA-ACL + RDF/OWL
Ethernet
Meeting Agent
Personal Agent
Presentation Rec. Agent
Lunch Agent
Emergency Alert Agent
Contexts in the Intelligent Spaces
Smart Tag Sensors
(Radio Frequency Identification)
Environment Sensors
(Xanboo & X10 technology)
Device & Gadget Sensors
(Java Ring, SmartCard etc.)

Pros/Cons of Broker

- Pros
  - supports resource-limited mobile computing devices
  - addresses the concerns for user privacy
- Cons
  - can create a “bottleneck” in large scale space
  - Solution: multiple brokers are grouped together to form a broker federation

COBRA-ONT

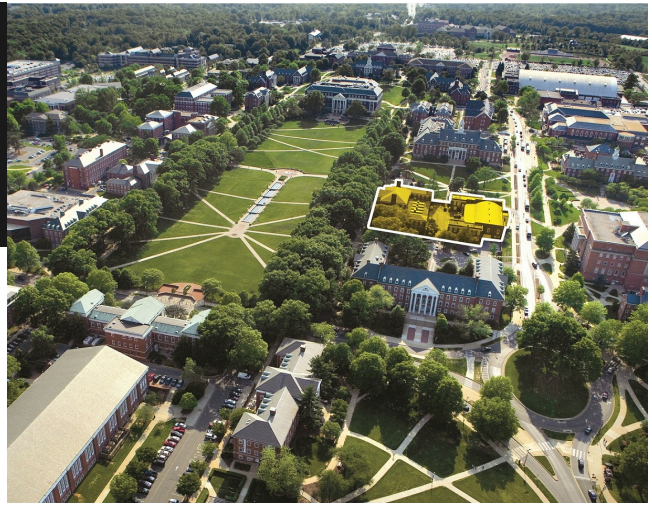
- ontologies expressed in the Web Ontology Language OWL
  - expressive ‘‘ontology language’’ by W3C
  - classes and properties
  - based on XML

An example

- a sensor detects Bluetooth cellphone in Rm 210
- sends event using COBRA-ONT to broker
- broker has identity of device
- Rm 210 is in CS building on campus
- broker concludes the person is in school today = context

CoBrA Ontology Classes		CoBrA Ontology Properties	
"Place" Related	Agents' Location Context	"Place" Related	Agent's Location Context
Place AtomicPlace CompoundPlace Campus Building AtomicPlaceInBuilding AtomicPlaceNotInBuilding Room Hallway Stairway OtherPlaceInBuilding Restroom Gender LadiesRoom MensRoom ParkingLot	ThingInBuilding SoftwareAgentInBuilding PersonInBuilding ThingNotInBuilding SoftwareAgentNotInBuilding PersonNotInBuilding	latitude longitude hasPrettyName isSpatiallySubsumedBy spatiallySubsumes accessRestricted-ToGender lotNumber	locatedIn locatedInAtomicPlace locatedInRoom locatedInRestroom locatedInParkingLot locatedinCompoundPlace locatedInBuilding locatedInCampus
"Agent" Related	Agent's Activity Context	"Agent" Related	Agent's Activity Context
Agent Person SoftwareAgent Role SpeakerRole AudienceRole IntentionalAction ActionFoundInPresntation	PresentationSchedule Event EventHappeningNow PresentationHappeningNow RoomHasPresentationHappeningNow ParticipantOfPresentation-HappeningNow SpeakerOfPresentationHappeningNow AudienceOfPresentationHappeningNow PersonFillsRoleInPresentation PersonFillsSpeakerRole PersonFillsAudienceRole	hasContactInformation hasFullName hasEmail hasHomePage hasAgentAddress fillsRole isFilledBy intendsToPerform desiresSomeone-ToAchieve	participatesIn startTime endTime Location hasEvent hasEventHappeningNow invitedSpeaker expectedAudience presentationTitle presentationAbstract presentation eventDescription eventSchedule

`Place` (class)



- properties
  - `longitude`, `latitude`, `name`
- subclasses
  - `AtomicPlace` and `CompoundPlace`
  - `CompoundPlace` subsumes `AtomicPlace`
  - Ex. Campus -> Building -> Room

OWL of Place

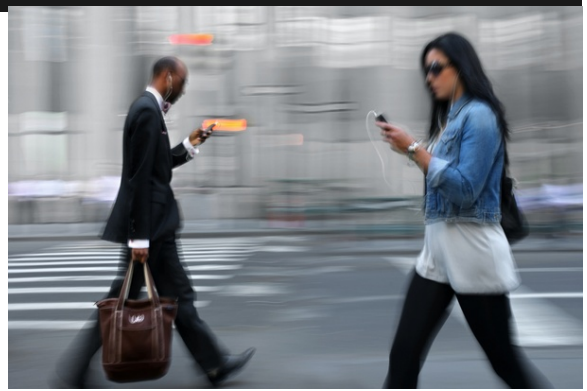
```
<owl:Class rdf:ID="Place">
  <owl:unionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#AtomicPlace"/>
    <owl:Class rdf:about="#CompoundPlace"/>
  </owl:unionOf>
  …
</owl:Class>

<owl:Class rdf:ID="AtomicPlace">
  <rdfs:subClassOf rdf:resource="#Place"/>
  <rdfs:subClassOf>

...
```

Agents

- **`Person`, `SoftwareAgent`**
- properties
  - `name`
  - `Role`
    - `SpeakerRole` and `AudienceRole`



Context of Agents

1. No agent can be physically present in two different atomic places during the same time interval
2. An agent can be physically present in two different compound places during the same time interval just in case one spatially subsumes the other

Event

- Presentation
  - invitedSpeaker and expectedAudience
  - RoomHasPresentationEventHappeningNow

COBRA-ONT

- just for meeting room domain
- need to be generalized for smart spaces

SOUPA: Standard Ontology for Ubiquitous and Pervasive Applications

Harry Chen, Filip Perich, Tim Finin, and Anupam Joshi

SOUPA Ontologies

- based on COBRA-ONT and many others
- intended to standardize ontologies
- two set of ontologies
  - **SOUPA Core** -> generic, universal
  - **SOUPA Extension** -> using core; specific

SOUPA Core

- person
- agent
- belief-desire-intention (BDI)
- action
- policy
- time
- space
- event

Person

- contact information and the profile of a person

```
<per:Person>
  <per:firstName
     rdf:datatype="&xsd;string>Harry</per:firstName>
  <per:lastName
     rdf:datatype="&xsd;string>Chen</per:lastName>
  <per:gender rdf:resource="&per;Male"/>
  <per:birthDate
     rdf:datatype="&xsd;date">1976-12-26</per:birthDate>
...
```

Policy

- set of rules that is specified by a user or a computing entity to restrict or guide the execution of actions
- **Security and Privacy**

Action

```
<owl:Class rdf:ID="ShareHarryLocInfoWithEBMembers">
    <owl:intersectionOf rdf:parseType="Collection">
       <owl:Class rdf:about="&act;Action"/>
```

...

- Policies permit or forbid actions

Agent & BDI

- Computing entities and humans
- BDI (beliefs, desires, intentions) of agent can give “mental state”
- `agt:believes`, `agt:desires`, and `agt:intends`
-

Time

- `tme:TimeInstant` and `tme:TimeInterval` classes

```
<tme:TimeInterval>
  <tme:from>
    <tme:TimeInstant>
      <tme:at rdf:datatype="xsd;dateTime">
        2004-02-01T12:01:01
      </tme:at>
    </tme:TimeInstant>
  </tme:from>
  <tme:to>
    <tme:TimeInstant>
```

…

- other properties: before, after, etc.

Space

- `spc:GeographicalSpace`
- `spc:GeographicalRegion`
- `spc:FixedStructure`
- `spc:FixedStructure`
- properties
  - coordinates, subsumes

Event

- Spatial and Temporal extensions
- Activities, schedules, sensing events

```
<owl:Class rdf:ID="DetectedBluetoothDev">
  <rdfs:subClassOf
   rdf:resource="&eve;TemporalSpatialEvent"/>
</owl:Class>

<owl:ObjectProperty rdf:ID="foundDevice">
  <rdfs:domain
  rdf:resource="#DetectedBluetoothDev"/>
</owl:ObjectProperty>

<DetectedBluetoothDev>
  <spc:hasCoordinates>
    <geo:LocationCoordinates>
      <geo:longitude rdf:datatype...>
```

SOUPA Extension

- define an extended set of ontologies to support specific domains
- demonstrate how to define new ontologies using SOUPA Core

SOUPA can be generalized

- Able to use/generalize SOUPA ontologies on CoBrA in a smart meeting room

Conclusions

- Ontologies are key for formalizing context
- SOUPA standardizes ontologies for context aware systems