A New Architecture for the Sensor Web:
The SWAP Framework

Deshendran Moodley
School of Computer Science,
University of KwaZulu-Natal, Durban, South Africa
moodleyd37@ukzn.ac.za

Ingo Simonis
Institute for Geoinformatics,
University of Muenster, Muenster, Germany
simonis@uni-muenster.de
ICT for Earth Observation Research Group
Meraka Institute, Pretoria, South Africa
isimonis@csir.co.za

UNIVERSITY OF KWAZULU-NATAL

meraka INSTITUTE
African Advanced Institute for Information & Communications Technology

What is the Sensor Web?

Open Geospatial Consortium (OGC)

".. *refers to web accessible sensor networks and archived sensor data that can be discovered and accessed using standard protocols and application programming interfaces*"

(Botts et. al. OGC 2006)

Our vision:

The Sensor Web is an infrastructure that allows end users to automatically access, and extract and use appropriate information from multiple sensor sources over the Internet

Outline

- Sensor Web Challenges
- OGC Sensor Web Enablement (SWE)
  - overview, challenges
- SWAP Approach
  - combine agents, ontologies
- SWAP Design
  - MASII, SWAP abstract architecture, ontologies
- Case study - wildfire detection
- Status - future work

Sensor Web: Challenges

Three key technical challenges:

1. *Publishing and discovering sensor resources*
   - create a publicly accessible infrastructure for publishing heterogeneous sensor resources and complex applications
   - discover and use sensor resources
2. *Sensor data fusion*
   - sensor data has different data models and formats and different spatial and temporal resolutions,
   - fusion -> higher spatial coverage and temporal resolution

Sensor Web challenges (cont.)

3. *Context-based information extraction*
   - end users have insufficient technical expertise and time to extract information from sensor data,
   - users require different views of the data according to needs/context
   - data can be filtered, summarised, transformed, or features can be extracted -> higher level features -> information -> application/decision making
   - same data can be reused for different applications

OGC Sensor Web Enablement (SWE)

- Open Geospatial Consortium (OGC) initiative
- *Web services*
  - **Sensor Observation service** for data access
  - **Sensor Planning Service** for sensor tasking and feasibility studies,
  - **Sensor Alert Service** for registering atomic conditions and push based notification,
  - **Web Notification Service** as a data transport protocol transformer
- *Data models and encodings*
  - **Observation & Measurement** (O&M)
  - **Sensor Model Language** (SensorML)

OGC Sensor Web Enablement (SWE)

- Attempts to address first two challenges
  - *publish* and *discover* sensor resources
  - data models and encoding facilitates *data fusion* but mostly a manual process
- No ontological infrastructure, lack of semantics restricts:
  - the discovery of data, automatic fusion of data and automatic processing of data
- Does not explicitly cater for data filtering and information overload.
- No methodology to build applications, application components are created and assembled in an adhoc manner
  - service providers *hide* complex *application logic* behind OGC web service interfaces
  - *resource intensive* to *create new views*, application components not publicly available, duplicated by service providers -> applications are created from scratch,
  - *views/applications are static*, cannot dynamically create new applications by reusing existing application components, users must interact with the original "raw" data when current views are not sufficient.

Technology paradigms

- **Multi-agent systems**:
  - publishing services over the Internet
  - discovering and invoking these services
  - ***knowledge level*** communication between systems -> eases interoperability
- **Supported by ontologies**:
  - semantic markup of service capabilities, data and tasks
  - semantic matching for matching services to tasks, aids in service discovery
  - agent communication, agent coordination, workflow specification/service composition
- **Integrate with OGC web services & SWE**

Sensor Web Agent Platform (SWAP)

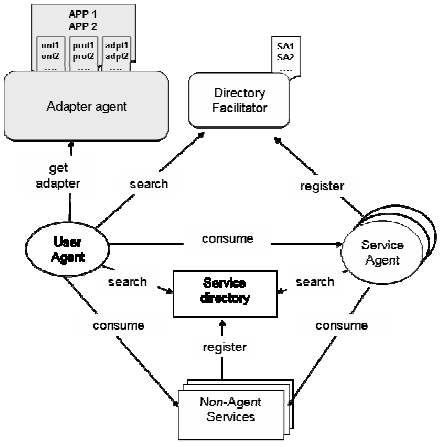
- Long term vision

Ontology driven information systems

MASII

- Multi-Agent System Infrastructure for the Internet
  - in-house research MAS platform (Java)
- *Design goals*
  - Application development and deployment
  - An application contains application components, i.e. user interfaces, application logic, ontologies, ACLs, protocols -> packaged to form an application adapter
  - Developers have flexibility to reuse application components or add new ones to form new application adapters

MASII



UNIVERSITY OF KWAZULU-NATAL

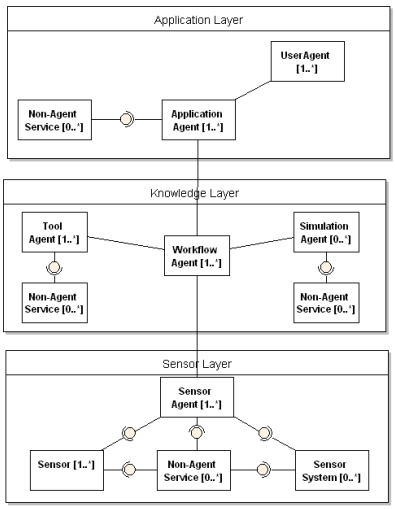
D. Moodley, I Simonis A New Architecture for the Sensor Web: The SWAP Framework

meraka INSTITUTE

SWAP Abstract Architecture

- Design goals:
  - easily expose new sensors and sensor data archives and to discover and access these resources
  - embed data transformation, data fusion and information extraction processes that can be applied to this data and reused for different applications
  - build applications for end users to access and respond to this information and data, by constructing and embedding automated rules in the system,
- Three layered abstract architecture
  - provides abstractions to reduce complexity for developing, deploying & maintaining applications
  - modular paradigm for developing and deploying Sensor Web applications -> promotes reusability

SWAP Abstract Architecture



UNIVERSITY OF KWAZULU-NATAL

D. Moodley, I Simonis A New Architecture for the Sensor Web: The SWAP Framework

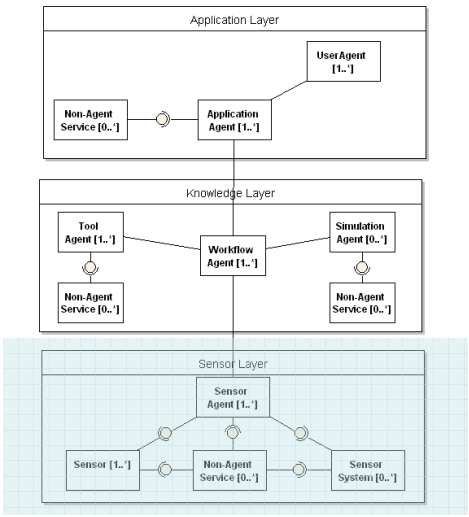
meraka INSTITUTE

SWAP: Sensor Layer

Sensor agents

- encapsulate individual sensors, sensor systems and archived observations
- expose sensor data in a uniform way and deal with any sensor-dependant processing
- data from sensor agents form input to agents in the second, *Knowledge Layer*

SWAP Abstract Architecture



UNIVERSITY OF KWAZULU-NATAL
D. Moodley, I Simonis A New Architecture for the Sensor Web: The SWAP Framework
meraka INSTITUTE

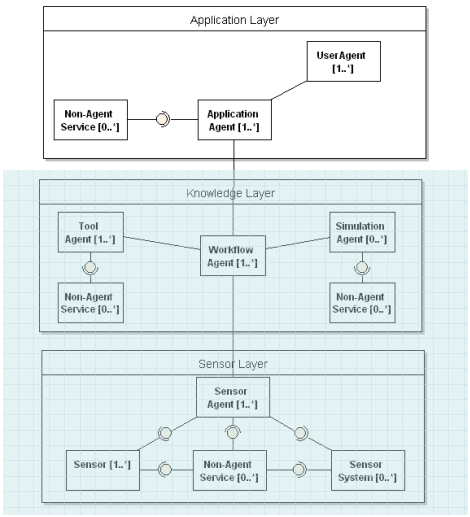
SWAP: Knowledge Layer

Tool agents
- provide feature extraction and image processing functionality

Simulation/Modeling agents
- store real-world models and can provide projections and analysis of data.

Workflow agents
- coordinate activities between agents using workflows/ service chains
- receive data from sensor agents and pass this data through a combination of tool and simulation agents and aggregate the results.
- the processed data stored by workflow agents typically form input to application agents in the *Application Layer*.

SWAP Abstract Architecture



UNIVERSITY OF KWAZULU-NATAL

D. Moodley, I Simonis A New Architecture for the Sensor Web: The SWAP Framework

meraka INSTITUTE

SWAP: Application Layer

Application agents:

- combines higher level features provided by workflow agents to form end user applications
- allows users to specify alerts in the system.
- provides different views of this data to different end users either via graphical user-interfaces for human users or an interface that can be invoked automatically by user agents.

User agents:

- software installed on a user's machine which automatically interacts with Sensor Web applications.
- contains an *application adapter* for each Sensor Web application used by the user.
- interacts automatically with application agents and integrates Sensor Web data/alerts into local applications and systems

Ontologies for data

- Inspired by NASA Sweet ontologies (top level ontology)
  - *concept* space for Earth system science
  - practical, engineering approach
  - provide support for representing time and space
  - physical measurable properties,

    e.g. air -> entity

    temperature -> property

    thus air pressure & air temperature can be represented as compound concepts i.e. the *temperature* of the *air, more scalable e.g. air pressure, water temperature, etc*
- *Our Approach*

  Sensor data is described in terms of the entity being observed, the physical property (of this entity) being measured, the time and the space over which it is measured and in terms of the larger data set to which this data belongs

Ontologies for agents

- ***Conceptual level***
  - agent must describe its data or service it provides and the spatiotemporal characteristics of this data without the implementation details
  - promotes conceptual/semantic interoperability
  - promotes dynamic extraction and integration of higher level features from sensor data
  - good conceptual description -> increases possibilities for reuse

Ontologies for agents

- ***Technical level***
  - agents must still exchange and process data
  - requires rich data types and data structures ranging from a single value at a specific time and space to multi-dimensional data over different spatial areas and varying time intervals
  - communication is by message passing, message structure is required
  - process flow or coordination between agents
- ***Mapping between levels***

  conceptual <-> technical <-> current programming languages, e.g. Java/C++

Agent communication

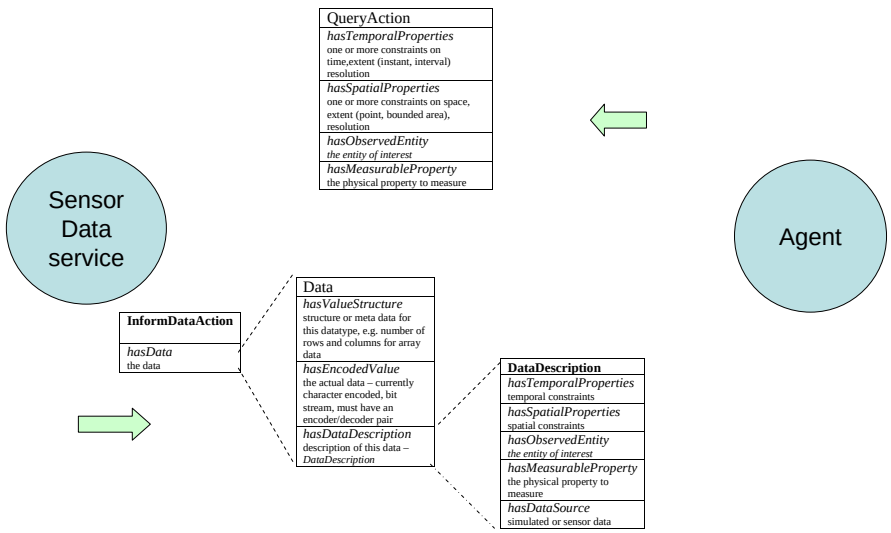
Agents communicate by sending messages to each other. Message must have *structure* and *semantics* so that they can be automatically composed by sending agents and automatically interpreted by receiving agents.

Consider a query for data action

Message structure

Message	
Sender	Agent that sent the message
Receiver	Agent to receive the message
Type	If this is an action message , then the type is the action type
Content	The query instance or result instance
Conversation ID	A unique conversation ID to store session information as interactions with different agents could occur at the same time.

Message Semantics – Data Query



UNIVERSITY OF KWAZULU-NATAL

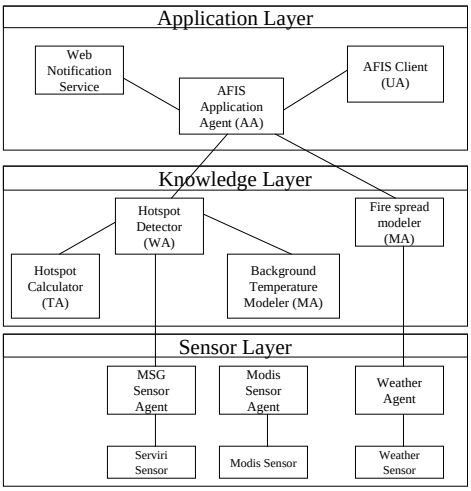
D. Moodley, I Simonis A New Architecture for the Sensor Web: The SWAP Framework

meraka INSTITUTE

Agent composition

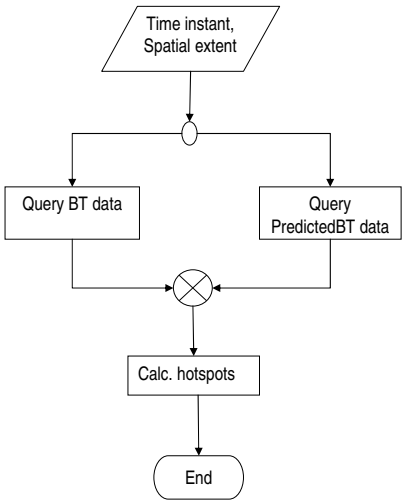
- OWL-S
  - extension of OWL for specifying processes, i.e. composite services
  - execution model is web services
- Not straightforward to use with agents
  - creating an agent to process mapping so that agent actions/messages can be mapped to OWL-S processes
  - specify processes with OWL-S, execute OWL-S processes using agents

Wildfire Detection



UNIVERSITY OF KWAZULU-NATAL
D. Moodley, I Simonis A New Architecture for the Sensor Web: The SWAP Framework
meraka INSTITUTE

Hotspot detection workflow



Status & outstanding work

- Completion of implementation of fire detection application
  - build a user agent and application agent, incorporate weather agent
- Ontological framework
  - *linking with SWEET* is currently very crude. Need to develop a tool and mechanism for easily linking with SWEET - better ontology construction tools are required
  - *data queries* are very simple at the moment.
    - requirements for a more formal query language, RDQL? to query for instances.
- Reasoning
  - currently very basic and hard-coded in Java.
  - alternative reasoning mechanisms being considered.
    - SWRL, Jess, Jena2, RuleML
    - **warning** the more technical the reasoning becomes, the harder it becomes for power users, domain experts and end-users to understand and maintain ontologies.

Status & outstanding work

- Agent based process or workflow specification and execution.
  - Testing agent-process-mapping OWL-S
- Second application
  - change detection in informal settlements

Thank you !

QUESTIONS?

UNIVERSITY OF KWAZULU-NATAL
D. Moodley, I Simonis A New Architecture for the Sensor Web: The SWAP Framework
meraka INSTITUTE