A Service-Oriented Multi-Agent Systems Architecture for the Sensor Web

A. Terhorst[1], I. Simonis[2] and D. Moodley[3] [1] Meraka Institute, CSIR, PO Box 395, 0001 Pretoria, South Africa, e-mail: aterhorst@meraka.org.za [2] Institute for Geoinformatics, University of Münster, 48149 Münster, Germany, e-mail: simonis@uni-münster.de
[3] School of Computer Science, University of Kwazulu Natal, 4041 Durban, South Africa, e-mail: moodleyd37@ukzn.ac.za

01 Introduction

Understanding the behaviour of ecosystems requires persistent and multi-modal observation. Advances in sensor technology and distributed computing, coupled with the development of open standards that facilitate sensor/sensor network interoperability, are contributing to the emergence of a phenomenon known as the 'Sensor Web'. This phenomenon can be described as an advanced Spatial Data Infrastructure (SDI) in which different sensors and sensor networks are combined to create a macro-instrument with massive sensing capability. The Sensor Web opens up avenues to fast assimilation of data from various sensors and to accurate analysis and informed decision making [1]. In this poster, we present a reference architecture for the Sensor Web that could serve as a potential candidate SDI for the South African Environmental Observation Network (SAEON). This architecture is a hybrid of the Foundation for Intelligent Physical Agents (FIPA) and Open Geospatial Consortium (OGC) standard architectures.

02 OGC Sensor Web Enablement

Sensor Web Enablement (SWE) is an OGC initiative that extends the OGC Web services framework[2] by providing additional services for integrating web-connected sensors and sensor systems. The SWE architecture is designed to enable the creation of web-accessible sensor assets through common interfaces and encodings[3]. Sensor assets may include the sensors themselves, observation archives, spatial databases, simulations, and algorithms for processing observations. Figure 1 illustrates the interaction between SWE web services.

Though promising, SWE has major drawbacks with respect to automation,scalability, interoperability and deployment and/or discovery of applications [4].

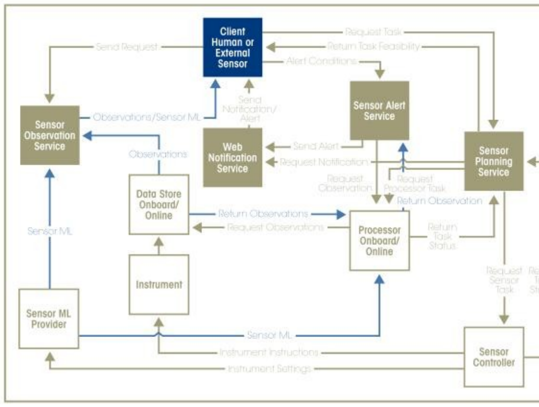

Figure1: SWE Service Interaction Diagram

03 Sensor Web Agent Platform

The Meraka Institute is developing a service-oriented multi-agent system architecture for the Sensor Web known as SWAP (Sensor Web Agent Platform). Our goal is to implement an adaptable sensor network that can reconfigure itself to answer user-defined queries about the state of the environment. SWAP incorporates the following key concepts into its architecture: ontologies, process models, choreography, directories and facilitators, service-Level agreements and quality of service measures[5]. The SWAP abstract architecture is split into three layers: Sensor Layer, Knowledge Layer and Application Layer (Figure 2). The Sensor Layer deals with components working directly on physical or virtual sensors in a tight or loose coupling. The Knowledge Layer contains re-usable simulation and processing components that can be dynamically assembled by Workflow Agents to deliver a variety of end-user applications. The Application Layer contains the user-interface that allows human users or other client machines to interact with the system.

04 Infrastructure Components

The abstract architecture is underpinned by the FIPA Agent Discovery Service Specification[6]. This specification describes standard infrastructure components that facilitate the discovery and consumption of agent and non-agent services. Figure 3 illustrates the agent/service framework used by SWAP: Applications register their capabilities with the Application Facilitator so that these may be discovered by User Agents. Likewise, Service Agents need to register their capabilities with the Directory Facilitator so that User Agents may discover and consume their services. Simililarly, Non-Agent Services register their capabilities with the Service Directory so that User Agents and Service Agents can discover and consume their services. User Agents are end-users that consume Non-Agent Services, Agent Services and Applications. User Agents connect to ontology servers, other agents and applications using adapters provided by Ontology/Adapter Agents. By integrating SWE services as Non-Agent Services, standardised OGC components can be incorporated into SWAP. Service Agents differ from Non-Agent Services in that they are more autonomous and able to communicate and reason with other agents.

05 Ontological Framework

An ontology is a formal explicit specification of a shared conceptualisation [7]. Conceptualisation refers to an abstract model of some phenomenon in which the relevant concepts of that phenomenon are described. Explicit means that the types of concepts used, and the constraints on their use are explicitly defined. Formal refers to the fact that the ontology should be machine-readable. Shared reflects the notion that an ontology captures consensual knowledge i.e. it is not private of some individual but accepted by a group [8]. Ontologies will provide explicit descriptions of all components within SWAP i.e. sensors and sensor data, simulation models, algorithms and applications, and how these components can be integrated and used by Workflow and User Agents. SWAP differentiates three levels of ontologies: upper ontologies, domain and generic task ontologies, and application ontologies (Figure 4). Developers of Sensor Web applications will typically use domain and generic task ontologies to specify new application ontologies.

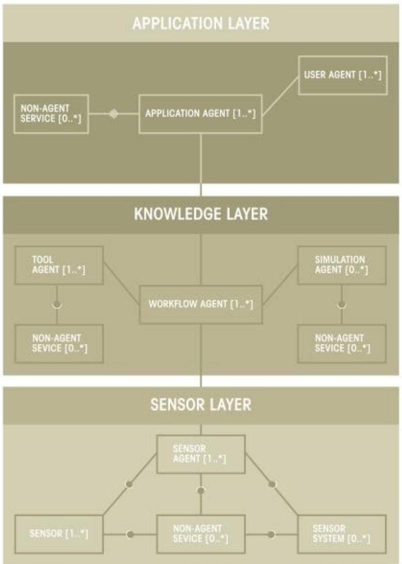

Figure 2: SWAP Abstract Architecture

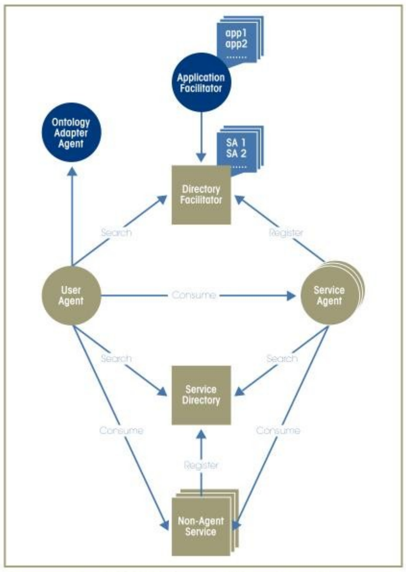

Figure 3: SWAP Agent/Service Framework

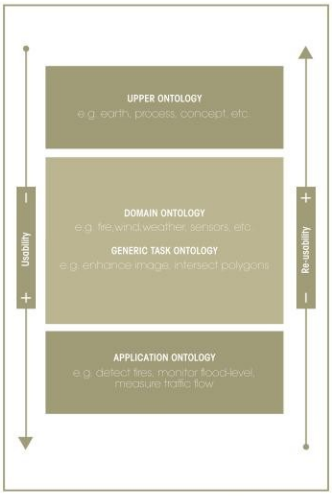

Figure 4: Ontology Levels

06 Use Case Scenario

We are presently developing a generic application that will allow ordinary users to set up custom sensor alert services without having to worry about any of the underlying sensor technology. Examples include a wildfire alert service, landcover change alert service, drought-onset alert service, flood warning alert service, and a traffic congestion alert service. For this to work, however, requires supporting domain and generic task ontologies. The Sequence Diagram presented in Figure 5 explains how the proposed application will work. Essentially, the application will provide users with an interface for creating an application/process ontology that drives the sensor alert workflow agent.

07 Conclusion

One of the major drawbacks of service-oriented SDI architectures is their passiveness. Services respond to requests sent by consumers but remain passive at all other times. SWAP will create a more active SDI that adapts to changes in its environment by integrating models and ideas produced by the AI community and it will allow domain experts and/or application providers to deploy new applications simply by formalising their knowledge as an ontology and making it available to the system.

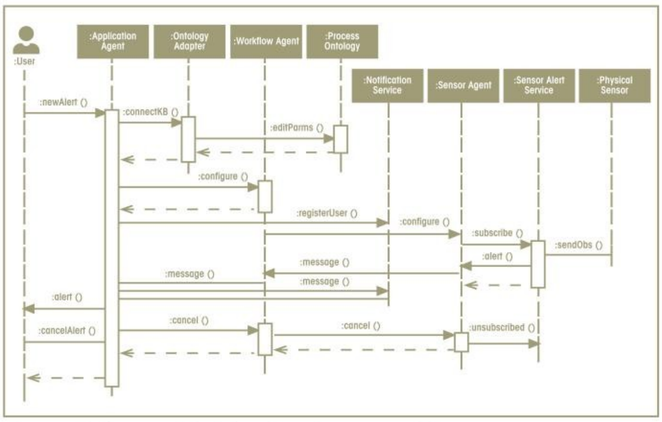

Figure 5: Sensor Alert Service Sequence Diagram

References [1] S. H. L. C. Liang and C. V. Tao, "A Distributed Geospatial Infrastructure for the Sensor Web" Computers & Geosciences, vol. 31, no. 2, pp. 221 - 231, 2005. [2] Anonymous, "The OGC Abstract Specification - Topic 0: Abstract Specification Overview" Document 04-084, Open Geospatial Consortium, Wayland, Massachussets, USA, 2005. [3] M. Reichardt, "Sensor Web Enablement White Paper" Document 05 063, Open Geospatial consortium, 2005. [4] D. Moodley and I. Simonis, "New Architecture for the Sensor Web: The SWAS Framework" Paper submitted to GI Science 2006 Conference. [5] M. R. Huhns, M. N. Singh and Others, "Research Directions for Service-Oriented Multiagent Systems" IEEE Internet Computing, pp. 65 - 70, November-December 2005. [6] Anonymous, "FIPA Agent Discovery Service Specification" Technical Specification PC00095A, Foundation for Intelligent Physical Agents, Geneva, Switzerland, 2003. [7] T. R. Gruber, Towards Principles for the Design of Ontologies Used for Knowledge Sharing. Deventer, The Netherlands: Kluwer Academic Publishers, 1993. [8] V. R. Struder, R. Benjamins and D. Fensel, "Knowledge Engineering: Principles and Methods" IEEE Transactions on Data and Knowledge Engineering, vol. 25, no. 1, pp. 161-197, 1998.

ifgi
Institute for Geoinformatics
University of Münster

meraka
INSTITUTE
African Advanced Institute for Information & Communications Technology

UNIVERSITY OF KWAZULU-NATAL