A Semantic Infrastructure for a Knowledge Driven Sensor Web

Deshendran Moodley[1,3] and Jules Raymond Tapamo[2,3]

[1]School of Computer Science, University of KwaZulu-Natal, Durban, South Africa
[2]School of Electrical, Electronic and Computer Engineering, University of KwaZulu-Natal, Durban, South Africa
[3]Centre for Artificial Intelligence Research, University of KwaZulu-Natal and CSIR Meraka, South Africa

Abstract. Sensor Web researchers are currently investigating middleware to aid in the dynamic discovery, integration and analysis of vast quantities of high quality, but distributed and heterogeneous earth observation data. Key challenges being investigated include dynamic data integration and analysis, service discovery and semantic interoperability. However, few efforts deal with the management of both knowledge and system dynamism. Two emerging technologies that have shown promise in dealing with these issues are ontologies and software agents. This paper introduces the idea and identifies key requirements for a Knowledge Driven Sensor Web and presents our efforts towards developing an associated semantic infrastructure within the Sensor Web Agent Platform.

Keywords: sensor web, ontologies, multi-agent systems, semantic middleware

1 Introduction

Advances in sensor technology and space science have resulted in the availability of vast quantities of high quality, but distributed and heterogeneous earth observation data. Sensor Web researchers are currently investigating middleware to facilitate the dynamic discovery, integration and analysis of this data with the vision of creating a global worldwide Sensor Web [33][6][9]. Key challenges being investigated include dynamic data discovery, integration and analysis, semantic interoperability, and sensor tasking. While it has been acknowledged that abstractions are required to bridge the gap between sensors and applications [6][9] and to provide support for the rapid deployment of end user applications [9], the most effective mechanism for modeling and managing the resultant deluge of software components remains an open issue. Two emerging technologies in Computer Science that have shown promise in dealing with these challenges are software agents and ontologies. Agent researchers propose the use of software agents as logical abstractions to model and manage software components in large scale, dynamic and open environments [17][34][35].

Software agents are autonomous software components that communicate at the knowledge level [13][17]. Many agent based architectures have been proposed for the Sensor Web [14][23][5][2]. However most approaches have limited support for the construction and evolution of the ontologies to support domain modeling, agent communication and reasoning, and to represent the algorithms, scientific theories and beliefs that are routinely applied to sensor data. In previous work we described an agent based architecture for the Sensor Web [21], i.e. the Sensor Web Agent Platform (SWAP), and proposed initial components for the semantic infrastructure [31]. In this paper we introduce the idea of a knowledge driven Sensor Web and describe a semantic infrastructure that supports both the specification and integration of scientific theories and system modeling. Additional details of the implementation of the ontologies and the reasoners can be found in [20].

The rest of the paper is organised as follows. In section 2 key requirements of a Knowledge Driven Sensor Web and its potential impact is described. Section 3 reviews related research. The SWAP semantic infrastructure is described in section 4 and in section 5 we conclude with a summary of key contributions and some avenues for future work.

2 A Knowledge Driven Sensor Web

A global Sensor Web must not only deal with issues around the provision, fusion and analysis of heterogeneous data. It must also support knowledge capture and use. Knowledge includes data processing and transformation algorithms, scientific theories and even subjective beliefs. To use this knowledge a mechanism must exist to dynamically apply knowledge to observations and to combine the results into meaningful information for end users. This capability to capture and apply knowledge will lead to a *Knowledge Driven Sensor Web (KDSW).*

A semantic infrastructure for a KDSW must include support for:

- Data and knowledge dynamism: a comprehensive but integrated conceptual modeling framework that includes support for not only modeling theme, time and space, but also uncertainty
- System and application dynamism: modeling of system entities, services, workflows, agents (system dynamism) and seamless movement between the conceptual model and the system model to support continuous application and service deployment

Potential benefits of a Knowledge Driven Sensor Web (KDSW) include [22]:

- Promoting the sharing and reuse of data, knowledge and services

- Facilitating human collaboration and scientific experimentation
- Reducing information overload and system complexity
- Managing both data, knowledge and system dynamism
- Increasing automation and machine intelligence

A Knowledge Driven Sensor Web can provide specific benefits to a wide range of users in the earth observation community. Decision makers can access, manage and visualise information provided by real time monitoring applications. Earth observation scientists can capture and share earth observation data and knowledge, and use the Sensor Web as a platform for experimentation, collaboration and knowledge discovery. Developers can easily design, develop and deploy dynamic Sensor Web services and end user applications.

3 Related work

A number of agent based Sensor Web approaches exist. These include the Internet-scale resource-intensive sensor network services (IrisNet) [14], Abacus [2], the agent based imagery and geospatial processing architecture (AIGA) [23], and the approach by Biswas et al. [5]. A summary of these approaches is given in [21]. Each approach proposes some form of layered architecture that provide abstractions to separate sensor agents from data analysis and filtering agents and aims to ease the modeling of agent based applications. While these approaches are promising for single or even groups of organizations building distributed agent based applications, except for the limited support provided in AIGA [23], no explicit support is provided for creating and managing ontologies that are required for agent communication and processing in an open Internet scale multi-agent system [13][34][35].

Ontologies are being widely investigated within the geospatial community to standardise, dynamically integrate and query complex earth observation data. Agarwal [1] summarises key advances in ontology research within the geospatial community. A more recent survey by Compton et. al. [8] describes the range and expressive power of twelve sensor ontologies. Despite these efforts there are still many outstanding challenges. The added temporal and spatial dimension associated with geospatial data requires additional representation support for modeling and formalising the domain [1][3]. One intuitive approach to model geospatial entities is to follow the human cognition system. Humans store knowledge in three separate cognitive subsystems within the mind [19]. The *what* system of knowledge operates by recognition, comparing evidence with a gradually accumulating store of known objects. The *where* system operates primarily by direct perception of scenes within the environment, picking up invariants from the rich flow of sensory information. The *when* system operates through the detection of change over time in both stored object and place knowledge, as well as sensory information. Separate ontological representations for space, time and theme have been proposed [26][31]. However, these approaches still lack support for representing the inherent uncertainty [3] associated with sensor data or for representing system entities. Even the widely used Web Ontology Language (OWL) [25] still lacks core support for representing time, space and uncertainty [30] and for representing system entities such as agents, services and processes.

4 The SWAP semantic infrastructure

Fig. 1 shows the different ontologies provided by SWAP. Ontologies are split into two levels, a conceptual level and a technical level. Conceptual ontologies are used for modeling and representing observations and theories about the physical world. Technical ontologies are used for modeling and representing the software entities (agents) that will host and process these observations and theories.

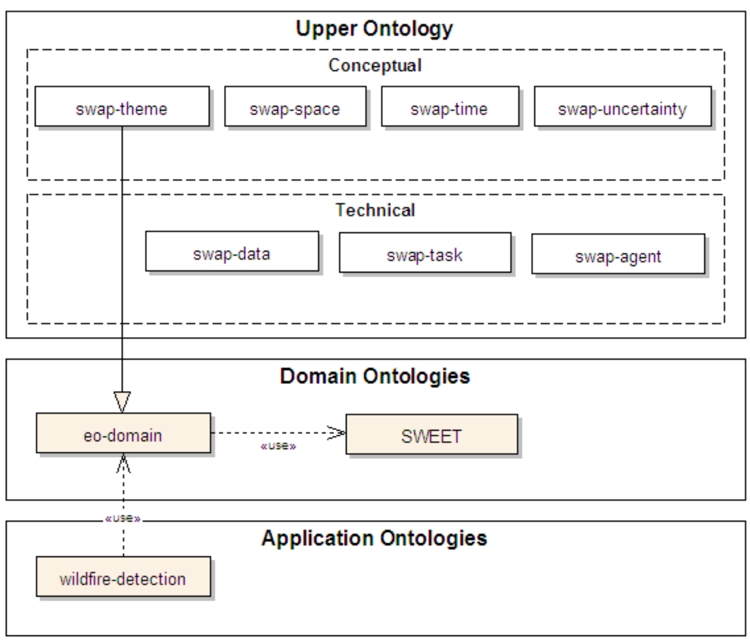

Fig. 1. SWAP ontology levels

The conceptual ontologies are based on creating separate subsystems as proposed by Mennis et al [19]. SWAP defines four conceptual dimensions to represent and reason about knowledge, the traditional dimensions of theme, space and time, and introduces a fourth dimension for uncertainty. An ontology and an associated reasoner is provided for each dimension. The reasoners currently use different inferencing engines: the thematic reasoner uses a Pellet reasoner; the temporal and spatial reasoners use a Jena rule-based engine; and the uncertainty reasoner uses a Bayesian inference engine. Domain ontologies for specific application domains are built by extending the *swap-theme* ontology. The *eo-domain* ontology extends the *swap-theme* ontology by adding concepts for building applications in the earth observation domain (Fig. 1). It currently references concepts from the SWEET [27] ontologies, an existing set of earth science ontologies. Application ontologies specify concepts that are used for specific applications, e.g. wildfire detection. Application specific concepts are specified along one or more of the four dimensions. The four reasoners are applied independently as required to perform inferencing on the application ontology.

4.1 The thematic dimension

The thematic dimension provides a thematic viewpoint for representing and reasoning about thematic concepts. The *swap-theme* ontology provides for the representation of observations and is based on the OGC's model of observations and measurements [10]. The *Observation* concept, defined in the *swap-theme* ontology, describes a single or a set of observations. Various thematic, spatial, temporal or uncertainty properties that are known may be specified for an observation (Fig. 2). The different types of properties are defined in the respective conceptual ontologies, e.g. thematic properties are defined in the *swap-theme* ontology and spatial properties are defined in the *swap-space* ontology.

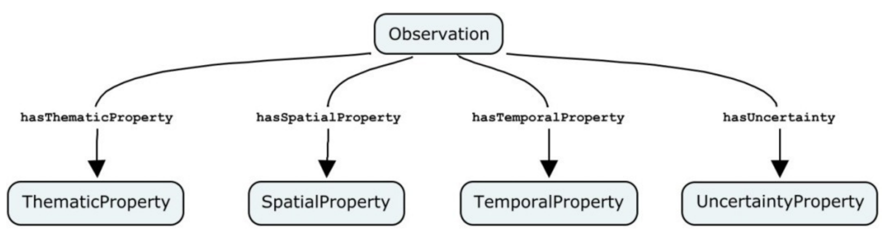

Fig. 2. Representing an observation

Two thematic properties are defined in *swap-theme*, *observesEntity* describes the entity being observed (*observedEntity*), while *observesProperty* describes the property of the entity that is being measured (*observedProperty*). The *eo-domain* ontology (Fig. 3) links observable properties from the NASA SWEET [27] property ontology by making these properties a subclass of *observedProperty* such as

BrightnessTemperature[1] and DryBulbTemperature[2]. Geographical entities from the SWEET earthrealm and SWEET phenomena ontologies are also linked by making these entities a subclass of *observedEntity*, e.g. *Air*, *Ocean*, *PlanetarySurface* and *Wind*.

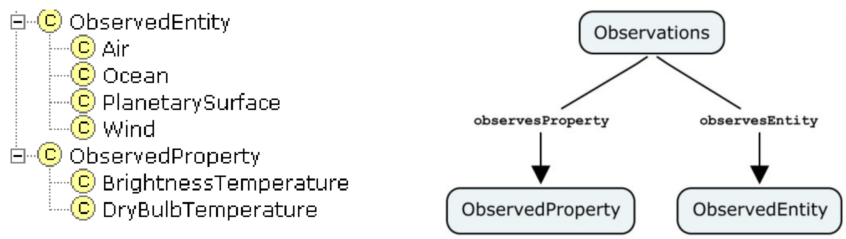

Fig. 3. The *eo-domain* ontology and representing a data set of observations

The schema for the thematic reasoner consists of the *eo-domain*, *swap-theme* and the SWEET ontology. This allows the inference engine to infer relations with SWEET concepts not explicitly referenced in the *eo-domain* ontology, e.g. that *BrightnessTemperature* and *DryBulbTemperature* are both subclasses of *Temperature*.

4.2 The spatial and temporal dimensions

The *swap-space* ontology provides concepts for representing and reasoning about the spatial aspects of data. A part of the *swap-space* ontology is shown in Fig. 4. Spatial entities include spatial reference systems, spatial projections, spatial resolution and location. Locations can be common descriptions such as a point coordinate or a bounding box, or well defined spatial geometries such as a point, line or polygon. A *SpatialThing* is defined as an entity that has a *Location* and the spatial reasoner determines how two *SpatialThings* are related. Since OWL does not provide native support for spatial representation, a set of spatial rules were formulated using the Jena[3] rule-based OWL reasoner to represent the eight spatial operators specified in the OpenGIS simple features for SQL [24].

---

[1] brightness temperature is the measure of the intensity of radiation thermally emitted by an object, given in units of temperature
[2] dry-bulb temperature is the temperature of air measured by a thermometer freely exposed to the air but shielded from radiation and moisture
[3] http://jena.sourceforge.net

For example, the rule used to determine whether two *SpatialThings* intersect is:

```
(?x spc:intersects ?y) <-
   (?x rdf:type spc:SpatialThing) (?y rdf:type spc:SpatialThing)
   (?x spc:locatedAt ?xExt) (?y spc:locatedAt ?yExt)
   spatiallyIntersects(?xExt,?yExt).
```

The rules use special builtins that were created for each of the eight relations. The builtins use the JTS topology suite [11] to determine if a specific relation holds between two spatial things. It first converts spatial things into JTS geometry objects and then calls the appropriate method on the geometry objects to perform the check.

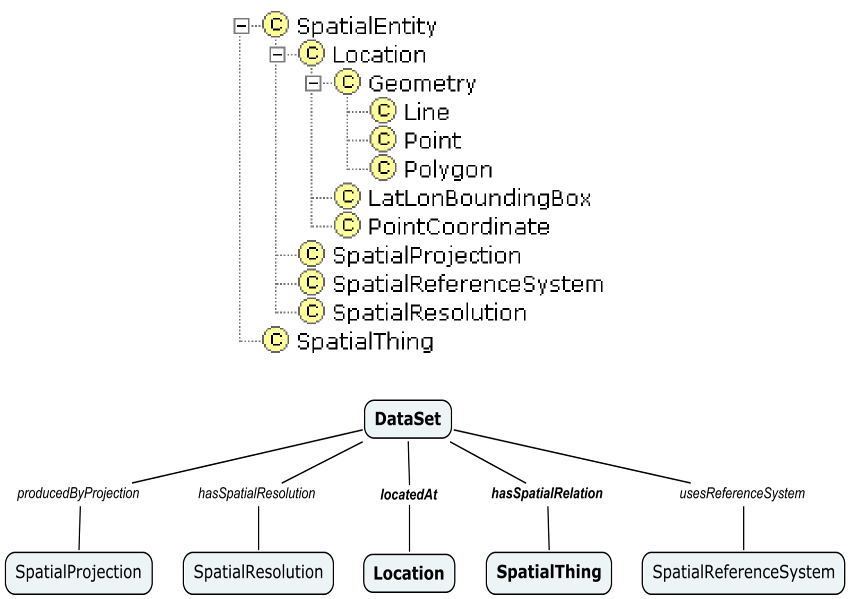

Fig. 4. The spatial ontology and representing spatial properties of observations in SWAP

The *swap-time* ontology incorporates the OWL-Time [16] ontology to represent and reason about the temporal aspects of data (Fig. 5). OWL-Time considers a temporal entity to be either a temporal instant or a temporal interval. As with the spatial reasoner an additional set of temporal rules, based on the COBRA temporal reasoner [7], specify temporal relations.

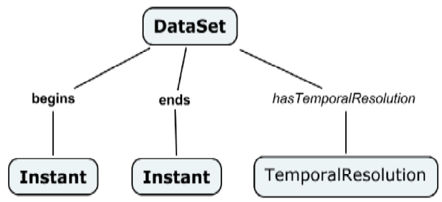

Fig. 5. Representing the temporal properties of a DataSet

For example, the two rules for determining whether a time instant is inside a time interval are:

```
 (?x tme:inside ?y) <-
    (?x rdf:type tme:InstantThing),
    (?y rdf:type tme:IntervalThing),
    (?y tme:begins ?beginsY), (?y tme:ends ?endsY),
    (?beginsY tme:before ?x), (?x tme:before ?endsY).

(?x tme:before ?y) <-
     (?x rdf:type tme:InstantThing),
     (?x tme:inCalendarClockDataType ?timeX),
     (?y rdf:type tme:InstantThing),
     (?y tme:inCalendarClockDataType ?timeY),
     lessThan(?timeX,?timeY).
```

where `tme` is the name space of the OWL-Time ontology. The first rule stipulates that a time instant x is within a time interval y if the starting time of y is before x, and x is before the ending time of y. The second rule uses the `lessThan` builtin to determine whether the time value of a time instant x is before the time value of another time instant y.

4.3 The uncertainty dimension

SWAP takes a Bayesian probability [28] approach to represent and reason about uncertainty on the Sensor Web. Bayesian probability is well suited for dealing with uncertainty on the Sensor Web: where no complete theory is available; where it exists it might be too tedious or complex to incorporate all the required observations; or where all the necessary observation data is not available [28].

The occurrence of natural phenomena is sometimes difficult to detect. However, certain phenomena sometimes exhibit consistent symptoms that are more easily detected and can serve as an indicator for the occurrence of the phenomena. The analysis of observations from multiple sensors may be required to determine the existence of the symptoms of specific phenomena. A Bayesian Network can be used to determine the probability of the occurrence of a phenomenon given one or more observable symptoms.

In such a Bayesian Network two types of discrete random variables are required:

- **Observable event variables:** represents the occurrence of a symptom of a phenomenon and is a qualitative measure for an observation. The variable must specify the entity, the characteristic of the entity being observed, as well as the property that contains the numerical value for the observation. The states are predefined numerical ranges, corresponding to qualitative descriptions. For example, wind speed is often used as an indication of the extent of a storm: from 6 to 49 km/hr is a breeze; 50 to 89 km/hr is a gale; 90 to 117 km/hr is a storm and speeds greater than 118 km/hr is indicative of a hurricane[4]. Observation values can be used to populate observable event variables.

- **Inferred event variables:** represents the occurrence of a phenomenon, e.g. a hurricane. A phenomenon is represented as a subclass of *Phenomenon* in the *swap-theme* ontology. When a phenomenon is detected, an instance of the appropriate class is created. These events are inferred from observable events or other inferred events. Even though these variables are intended for representing the occurrence of a phenomenon, they can be used to represent any event that is not easily or directly measurable.

An occurrence of an observable event is determined by evaluating measurements of some observed property of an observed entity, e.g. the speed of the wind above a certain threshold results in the occurrence of a "strong wind" event. These observable events are used to infer the probability of the occurrence of other events, e.g. a very strong wind is a symptom of a hurricane event. Thus, by analysing one or more measurements certain phenomena can be detected, e.g. a wind speed above 118 km/hr and an air pressure lower than 97.7 kPa can be considered to be symptoms of a hurricane event[5]. A simple Bayesian Network for determining the probability that a hurricane is occurring is shown in Fig. 6. The proposed Bayesian Network model assumes that all variables are discrete and represent events that occur at the same time and space. A limitation of the current model is that it does not cater for the influence of past or future events, or the influence of events occurring at different locations.

An ontology to represent Bayesian Networks.
The *swap-uncertainty* ontology, shown in Fig. 7 extends the BayesOWL [12] ontology. The BayesOWL ontology proposes five classes to represent a Bayesian Network, i.e. *ProbObj*, which could either be a *CondProb* or a *PriorProb*, *Variable* and *State*. A *ProbObj* has a probability value (*hasProbValue*) of some variable

[4] Using the Beaufort scale from http://www.hwn.org/home/bws.html
[5] Using the Saffir-Simpson Hurricane Wind Scale, from http://www.nhc.noaa.gov/sshws.shtml

(*hasVariable*) being true. In BayesOWL a *Variable* represents whether an instance is a member (*rdf:type*) of the specified class (*hasClass*) with one of two states, either *True* or *False*.

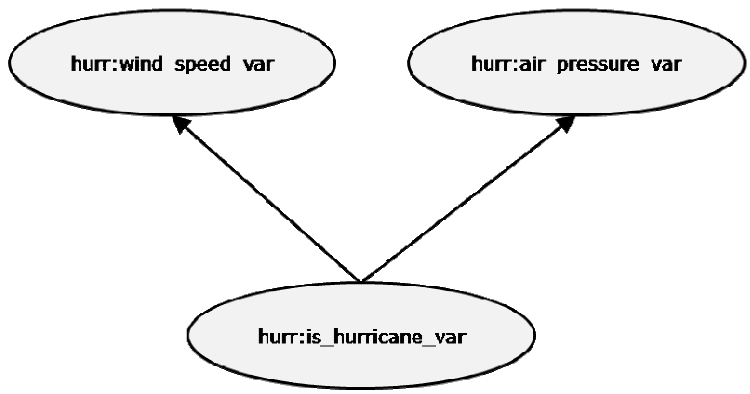

Fig. 6. A Bayesian Network to determine the occurrence of an hurricane from air pressure and wind speed observations

One extension to the BayesOWL ontology is the specialization of the *State* class to allow for user defined *DiscreteStates*. The *DiscreteRangeState* could be a numeric interval for numerical data type properties or a *SingleNumericState* for single numeric values.

The *swap-uncertainty* ontology provides support to represent one or more Bayesian Networks (BN). Each node in the BN represents either an observation or an inferred variable. An observation variable represents the observation value (*hasValueProperty*) for some observed property (*observesProperty*) of the observed entity (*observesEntity*). An inferred variable represents the occurrence of some phenomena (*hasClass*). The *influencedBy* property is used to specify the variables that influence the state of the variable.

SWAP uses the BNJ toolkit for internal representation and inferencing. Bayesian Network tools in Java (BNJ)[6]. BNJ is an open source Java toolkit for developing applications that use Bayesian Networks. It provides a visual Bayesian Network editor and viewer, a graph representation model for representing and manipulating a

---

[6] http://bnj.sourceforge.net

BN, a number of inference engines, as well as learning algorithms for constructing a Bayesian Network from data.

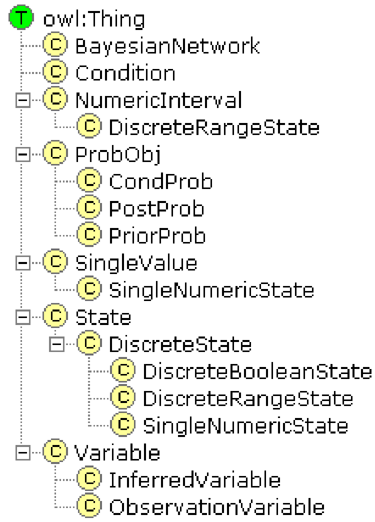

Fig. 7. A fragment of the SWAP uncertainty ontology

A *BayesianNetwork* instance uses the states of observed variables (observation instances) to make inferences about whether a phenomena has occurred (inferred variables). If a phenomena has occurred then an instance of the corresponding phenomena, which contains the corresponding location and time of the observations, is created. A schema ontology containing the BN and observation instances from the knowledge base are provided to the inference engine. The BN is first extracted from the schema ontology and used to create a BNJ graph model. The URIs of the variables and their states are used as the variable and state names in the BNJ graph model to ease the mapping of variables and states between the ontology and the graph model.

In this way the uncertainty reasoner dynamically populates user defined bayesian networks with observable events, performs inferencing on these events and determines and records the occurrence of other events.

4.4 System ontologies

SWAP provides three technical ontologies, i.e. *swap-data*, *swap-agent* and *swap-task* that provide representational support to describe the system entities that are required for hosting and transmitting observations, and for executing algorithms and theories.

The *swap-data* ontology provides descriptions of different data structures that can be exchanged between agents. This includes coverage (image) and feature data as well as units of measure.

Representing agents

The *swap-agent* ontology provides support for representing an agent, the service it hosts and the interaction protocol required to invoke the service. It provides support for representing the six different types of agents specified in the SWAP abstract architecture (Fig. 8) [21]. These are data provider (Sensor) agents, processing or data transformation (Tool) agents, modeling (Modeling) agents and coordination (Workflow) and application (Application) agents.

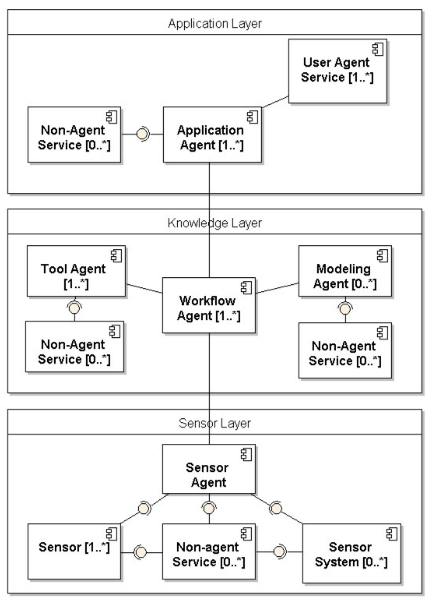

Fig. 8. The SWAP abstract agent architecture [21]

Each agent type has a corresponding service description with a set of common attributes that capture the conceptual functionality of the service. Sensor Agents provide a description of the observations that they provide, while Tool and Modeling Agents provide a description of the data processing algorithms and prediction models that they respectively provide. Service description attributes are grouped into the four different conceptual systems, i.e. spatial, temporal, thematic and uncertainty, and are specified using concepts from the appropriate top level ontology. Service descriptions also contain service invocation information in the form of input and output mappings.

A request and a response message template is used for invoking and interpreting the response of the service. The request message template specifies all service invocation parameters, which may be mandatory or optional parameters that have default values. Users populate mandatory parameters and may also specify optional parameters for finer control of the service. These message templates are used to dynamically invoke a service and to consume and interpret its results. This bridges the gap between service selection and use, i.e. once a suitable service has been identified it can be dynamically invoked and its results can be dynamically interpreted.

Representing services and workflows

The *swap-task* ontology is based on OWL-S [32], and provides algorithmic primitives to assemble multiple agents into executable agent workflows. An agent is represented as atomic processes and OWL-S algorithmic constructs are used to assemble multiple agents into appropriate sequences of invocations or composite processes. The main extension to OWL-S is a process to agent mapping that allows OWL-S processing steps to be transformed into agent invocations at runtime. The mapping specifies request and response templates that are used to transform each processing step into an appropriate request and response message used to invoke an agent and to interpret its response.

The technical ontologies provide support for describing the services offered by different agents and the agent interactions used to invoke these services. Support is also provided for constructing complex information processing chains or workflows that may be stored, shared and executed on demand. Since service descriptions and data models are captured within shared ontologies, they become dynamic entities that can be accessed, queried and modified at runtime. Selected services can be assembled into different configurations to form complex executable workflows that may be deployed as new composite services. This approach facilitates interoperability between agents, and between agents and humans. It also allows for data models and service offerings to change, and evolve naturally with minimal impact and without having to re-engineer the system.

Together, the technical and conceptual ontologies allow SWAP users to represent complex information processing chains or workflows. Users search semantic agent service descriptions and identify appropriate sensor data sets, algorithms and models to apply to these data sets. Once the appropriate agents are identified, users use the algorithmic constructs in the *swap-task* ontology to specify a processing workflow that assembles different agent services in an appropriate sequence for execution. Each workflow represents new functionality in the system. A workflow can also be deployed on a Workflow Agent where it can be executed on demand. Since a workflow is fully specified and executed from its OWL-S specification, the appropriate ontologies (which contain the workflow) can be shared, downloaded and executed locally. Furthermore, once the workflow is downloaded it can be easily modified and executed locally by SWAP users. A workflow is represented as a composite process, which means that it can be incorporated into other composite processes (workflows). This allows for reuse of existing workflows within other workflows and for creating and managing large and complex nested workflows. Currently, workflows are created and modified manually via an ontology editor. However, given that the semantics of both the conceptual and the technical aspects of each service are specified in the service description, this provides a sound foundation for automating workflow composition.

5 Conclusion

We have introduced the notion and proposed knowledge representation requirements and potential benefits of a Knowledge Driven Sensor Web. We contend that a semantic infrastructure and formal software modeling and engineering abstractions are both equally important to manage data and knowledge dynamisms as well as system and application dynamism. We propose an ontology driven multi-agent system approach to constructing such a system. A key limitation in agent based approaches is the lack of a comprehensive semantic infrastructure that includes support for representing uncertainty, theories and beliefs and support for representing agents, services and tasks. A semantic infrastructure that deals with these limitations was described. A novel aspect is the introduction of the additional modeling dimension of uncertainty which can be used for representing and applying subjective theories. The *swap-uncertainty* ontology incorporates Bayesian probability, which is widely used in practical applications to represent degrees of belief, and allows for the incorporation of Bayesian Networks to represent different theories of cause and effect relations between events in the physical world. The nature and availability of sensor data, the accuracy and completeness of the theory that underpins the choice, and the sequence of the processing steps may contribute an additional element of uncertainty. The information produced by workflows is frequently approximations or best guesses. The incorporation of uncertainty allows end users to better understand the quality of information generated within the Sensor Web.

There are many avenues for future work. The relation of this work to the trend in the Semantic Sensor Web community towards linked data [9][4][18] warrants further investigation. Another avenue is the extension of the uncertainty model to capture and reason about relations between past, current and future events and events occurring at different locations.

REFERENCES

1. Agarwal, P. (2005), 'Ontological considerations in GIScience', International Journal of Geographical Information Science 19(5), 501-535.
2. Athanasiadis, I.N.; Milis, M.; Mitkas, P.A. & Michaelides, S.C. (2005), Abacus: A multi-agent system for meteorological radar data management and decision support, in 'ISESS-05: International Symposium on Environmental Software Systems, Sesimbra, Portugal, May 2005'.
3. Balazinska, M.; Deshpande, A.; Franklin, M.J.; Gibbons, P.B.; Gray, J.; Hansen, M.; Liebhold, M.; Nath, S.; Szalay, A. & Tao, V. (2007), 'Data Management in the Worldwide Sensor Web', IEEE Pervasive Computing 6(2), 30-40.
4. Barnaghi, P. & Presser, M. (2010), Publishing Linked Sensor Data, in ‘Kerry Taylor, Arun Ayyagari, David De Roure (Eds.): The 3rd International workshop on Semantic Sensor Networks 2010 (SSN10) in conjunction with the 9th International Semantic Web Conference (ISWC 2010), 7-11 November 2010, Shanghai, China'.
5. Biswas, P.K.; Schmiedekamp, M. & Phoha, S. (2006), 'An agent-oriented information processing architecture for sensor network applications', Int. J. Ad Hoc Ubiquitous Comput. 1(3), 110-125.
6. Bröring, A.; Echterhoff, J.; Jirka, S.; Simonis, I.; Everding, T.; Stasch, C.; Liang, S. & Lemmens, R. (2011), 'New Generation Sensor Web Enablement', Sensors 11(3), 2652-2699.
7. Chen, H.L. (2004),'An Intelligent Broker Architecture For Pervasive Context-Aware Systems', PhD thesis, University of Maryland, Baltimore County.
8. Compton, M.; Henson, C.; Neuhaus, H.; Lefort, L. & Sheth, A. (2009),A Survey of the Semantic Specification of Sensors, in 'Kerry Taylor, Arun Ayyagari, David De Roure (Eds.): The Proceedings of the 2nd International Workshop on Semantic Sensor Networks (SSN09), in conjunction with the 8th International Semantic Web Conference (ISWC 2009), Washington DC, USA, October 26, 2009'.
9. Corcho, O. & Garcia-Castro, R. (2010), 'Five challenges for the Semantic Sensor Web', Semantic Web 1(1-2), 121-125.
10. Cox, S. (2006),'Observations and Measurements. Discussion Paper. OGC 05-087r3 Version 0.13.0 Observations and Measurements'.
11. Davis, M. (2007), Secrets of the JTS Topology Suite, ‘Free and Open Source Software for Geospatial 2007 (FOSS4G2007), Victoria, Canada, 24-27 September 2007'.
12. Ding, Z.; Peng, Y. & Pan, R. (2006), Soft Computing in Ontologies and Semantic Web, Springer-Verlag, chapter BayesOWL: Uncertainty Modeling in Semantic Web Ontologies, pp. 3-29.
13. Gaspari, M. (1998), 'Concurrency and knowledge-level communication in agent languages', Artif. Intell. 105(1-2), 1-45.
14. Gibbons, P.B.; Karp, B.; Ke, Y.; Nath, S. & Seshan, S. (2003), 'IrisNet: An Architecture for a Worldwide Sensor Web', IEEE Pervasive Computing 2(4), 22-33.
15. Hakimpour, F.; Aleman-Meza, B.; Perry, M. & Sheth, A. (2006), Data Processing in Space, Time and Semantics Dimensions, in ‘Terra Cognita Workshop - Directions to the Geospatial Semantic Web, A workshop of the 5th International Semantic Web Conference ISWC 2006, November 5-9, 2006, Athens, Georgia, USA'.
16. Hobbs, J.R. & Pan, F. (2004), 'An ontology of time for the semantic web', ACM Transactions on Asian Language Information Processing (TALIP) 3(1), 66-85.

17. Jennings, N.R. (2001), 'An agent-based approach for building complex software systems', Commun. ACM 44(4), 35-41.
18. Kessler, C. & Janowicz, K. (2010), Linking Sensor Data - Why, to What, and How?, in ‘Kerry Taylor, Arun Ayyagari, David De Roure (Eds.): The 3rd International workshop on Semantic Sensor Networks 2010 (SSN10) in conjunction with the 9th International Semantic Web Conference (ISWC 2010), 7-11 November 2010, Shanghai,China'.
19. Mennis, J.; Peuquet, D. & Qian, L. (2000), 'A conceptual framework for incorporating cognitive principles into geographical database representation', International Journal of Geographical Information Science 1414(6), 501-520.
20. Moodley, D. (2009),'Ontology Driven Multi-Agent Systems: An architecture for Sensor Web applications', PhD thesis, School of Computer Science, University of KwaZulu-Natal, Durban, South Africa.
21. Moodley, D., and Simonis, I. (2006), A new architecture for the sensor web: the SWAP-framework, in ‘Semantic Sensor Networks Workshop, A workshop of the 5th International Semantic Web Conference ISWC 2006, November 5-9, 2006, Athens, Georgia, USA.
22. Moodley, D.; Vahed, A.; Simonis, I.; McFerren, G. & Zyl, T.V. (2008), 'Enabling a new era of Earth observation research: scientific workflows for the Sensor Web', Ecological Circuits 1, 20-23.
23. Nolan, J. (2003),'An agent-based architecture for distributed imagery and geospatial computing', PhD thesis, George Mason University.
24. Open GIS Consortium, I. (May 5, 1999),'OpenGIS Simple Features Specification For SQL Revision 1.1', Technical report, OpenGIS Project Document 99-049.
25. Patel-Schneider, P.F. & Horrocks, I. (2007),'OWL 1.1: Web Ontology Language Overview'.
26. Perry, M.; Hakimpour, F. & Sheth, A. (2006),Analyzing theme, space, and time: an ontology-based approach, in 'GIS '06: Proc. 14th annual ACM international symposium on Advances in geographic information systems', ACM Press, New York, NY, USA, pp. 147-154.
27. Raskin, R.G. & Pan, M.J. (2005), 'Knowledge representation in the semantic web for Earth and environmental terminology (SWEET)', Computers & Geosciences 31(9), 1119-1125.
28. Russell, S. & Norvig, P. (2003), Artificial Intelligence: A Modern Approach, Prentice-Hall, Englewood Cliffs, NJ.
29. Russomanno, D.J.; Kothari, C.R. & Thomas, O.A. (2005), Building a Sensor Ontology: A Practical Approach Leveraging ISO and OGC Models, in 'IC-AI', pp. 637-643.
30. Shadbolt, N.; Berners-Lee, T. & Hall, W. (2006), 'The Semantic Web Revisited', IEEE Intelligent Systems 21(3), 96-101.
31. Terhorst, A.; Moodley, D.; Simonis, I.; Frost, P.; McFerren, G.; Roos, S. & van den Bergh, F. (2008), Geosensor Networks, Lecture Notes in Computer Science, Volume 4540/2008, Springer-Verlag, chapter Using the Sensor Web to Detect and Monitor the Spread of Vegetation Fires in Southern Africa, pp. 239-251.
32. The OWL-S Coalition (2004),'OWL-S: Semantic Markup for Web Services'. Available online at: http://www.w3.org/Submission/OWL-S/
33. van Zyl, T.L.; Simonis, I. & McFerren, G. (2009), ''The Sensor Web: systems of sensor systems', International Journal of Digital Earth 2:1, 16-30.
34. Wijngaards, N.J.E.; Overeinder, B.J.; van Steen, M. & Brazier, F.M.T. (2002), 'Supporting internet-scale multi-agent systems', Data Knowl. Eng. 41(2-3), 229-245.
35. Zambonelli, F.; Jenning, N.R.; Omicini, A. & Wooldridge, M.J. (2001), Agent-oriented software engineering for Internet agents, Springer-Verlag, London, UK, pp. 326-346.