Sensor Ontologies: From Shallow to Deep Models

David J. Russomanno, Cartik Kothari, and Omoju Thomas
Electrical and Computer Engineering Department
The University of Memphis
Memphis, TN 38152 USA
{d-russomanno, rkothari, oathomas}@memphis.edu

Key Words: Sensor Ontology, Sensor Fusion, Semantic Web

***Abstract*—This paper presents a practical approach to developing comprehensive sensor ontologies based upon deep knowledge models rather than capturing only superficial sensor attributes. It is proposed that the representation and utilization of deep sensor ontologies will enable a variety of sensor information system applications including sensor parts compatibility determination, dynamic sensor selection and tasking, and reasoning about systems of sensors in which data must be fused and queried from a variety of sensor types within a myriad of environments.**

I. INTRODUCTION

Advances in sensor technology are requiring new strategies and tools for how data is collected, shared, integrated, and used in making decisions. There are several rapidly evolving efforts related to fundamental sensor research enabling the design of increasingly complex sensor-based systems. Many of these efforts focus on smart materials technology made possible by advances in semiconductor processes and structures including large-scale-manufacturing of interconnections, micro-electronic circuits, and micromachining [1].

In parallel to advances in core sensor technology, recent advances in wireless technology have enabled the dynamic integration of large-scale sensor systems that support real-time spatial and temporal sampling and decision-making in rich tactical environments [2]. Furthermore, advances in knowledge representation standards, reasoning techniques, and service specification standards, including evolving Semantic Web infrastructure [3], have provided the organizational framework for the unambiguous exchange of knowledge and data and the determination of services within a networked environment, which will ultimately facilitate the delegation of significant sensor-related tasks to intelligent, software agents.

Not withstanding these developments, the *Committee on New Sensor Technologies: Materials and Applications* has implied that the potential of advanced sensor technology has been limited by the lack of a well-accepted and common language for expressing sensor definitions, attributes, classifications (including taxonomies), as well as descriptions of sensor needs and performance [1]. Although this committee documented an approach that exploits a common framework for describing sensors and sensing-system applications, the strategy is not based upon a formal conceptualization and it is not expressed in a computer-readable format.

The need for sensor ontologies to establish a widely-accepted terminology of sensors, their properties and capabilities has recently appeared in the literature [4-5] to enhance fusion and interoperability in a network-centric environment. However, current computer models of sensor ontologies are nonexistent or tend to be shallow, with only superficial aspects of sensors expressed in taxonomies captured as class hierarchies.

As part of the scope for the Center for Advanced Sensors at The University of Memphis we are attempting to develop deeper models in which a sensor ontology consists of a formal conceptualization of sensors including their categories, taxonomy, relationships, and meta data regarding sensor characteristics, performance, and reliability. Furthermore, logical statements that describe associations among sensor concepts, as well as aspects of sensor operating principles, computing capabilities, communications capabilities, and other pertinent semantic content in a machine-readable format are to be included in a deep, comprehensive sensor ontology. Part of this effort requires an analysis of traditional, physics-based models of sensors, which are often developed in Matlab, to determine what aspects of these models are appropriate for inclusion into an ontology.

In this paper, we present our pragmatic approach for the construction of a sensor ontology. Our approach includes deployment of some currently available techniques and tools based upon emerging Semantic Web infrastructure such that software agents can interpret and reason with comprehensive sensor knowledge bases, that is, instances of ontologies loaded into a computational environment.

Although we make no claims that an orthogonal, complete or universally acceptable sensor ontology is feasible, we provide pragmatic examples, with supporting rationale and using currently available tools, that we propose will lead toward the deployment of widely-used sensor ontologies in a variety of application domains.

II. SENSOR ONTOLOGY

A. Basic Ontology

The term ontology has been used in a variety of contexts. For our purposes, we adopt a conceptualization of declarative knowledge as described by Genesereth and Nilsson [6] to define an ontology and it includes, but is not necessarily limited to, the following:

- the classes to which objects belong (e.g., sensor types)
- the class hierarchy or taxonomic structure (e.g., set of radiant sensors is a subset of all sensors)
- the relational basis set among the classes (e.g., a sensing elements is *part of* a sensor)
- the functional basis set among the objects (e.g., bandwidth('JERS SAR') = 1.275 GHz)
- the capability for executing special programs or procedures for evaluating the truth of literals or attribute values (e.g., procedural attachment)

Once a basic ontology has been defined, a language, which can be unambiguously interpreted by a computer, can be used to express knowledge using concepts defined in the ontology. A knowledge base is created by instantiating a set of objects under study (e.g., specific sensor instances).

Consider the development of a basic sensor ontology. Since there will be different user communities, a taxonomy of sensor types could be based upon a variety of classification methods to impose an order on the hierarchy. Several possible classification dimensions are possible, including, but not limited to, those shown in Fig. 1.

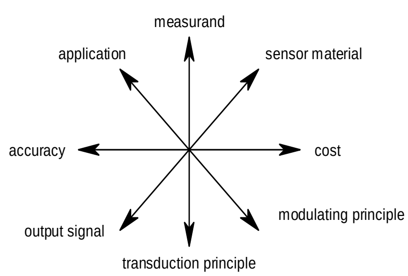

Fig. 1: Representative Sensor Classification Dimensions

For example, the Unified Modeling Language (UML) class model in Fig. 2 has been developed to aid in the visualization of a skeletal sensor taxonomy (note the sensor class's discriminator attribute *measurand* has been used here to impose the order upon the sensor hierarchy). UML class diagrams are good for illustrating aspects of an ontology (in particular, the taxonomic hierarchy of the classes in the ontology and the associations among the classes). Converting UML diagrams to an ontology representation language, such as the Web Ontology Language (OWL) [7], requires some judgment since there is not a one-to-one mapping between the two modeling formalisms and UML alone does not adequately model all aspects of an ontology.

Fig. 2 includes sensor definitions that have been adopted based upon the recommendations of the *Committee on New Sensor Technologies: Materials and Applications* [1]:

Sensor Element: the fundamental transduction mechanism (e.g., material) that converts one energy form to another

Sensor: a sensor element including its physical and external connections

Sensor System: a sensor and its associated signal processing hardware

The UML model shown in Fig. 2 conceptualizes a possible sensor system that is made up of several devices including a sensor. In the model, a sensor is a type of device, which can have several connection nodes. Each connection node may be involved in up to one actual connection in a physical sensor system. Several devices could be connected at a give node thereby supporting the modeling of connectivity of devices (sensor, modulating device, power supply, analog or digital filters, etc.) which comprise a sensor system. The model permits a lattice structure to support multiple inheritance (for example, a multi-band sensor that will inherit attributes, associations, and capabilities from many parents). Classification of sensors below the sub classes formed by discriminating on attribute *measurand* may use other dimensions. For example, the focusing technique may form the basis of the SAR class while the sensing band may form the basis of the photoconductive class.

B. Comprehensive Ontology

A comprehensive ontology must also include a domain theory expressed in a language constructed using the functional and relational basis sets to support ontology-driven inference. For example, smart sensors may embed computing and communication capabilities within their package. A comprehensive sensor ontology must capture these capabilities in the representation, as well as the sensor percept attributes, that is, the attributes needed to store measurements of physical phenomena and detection, classification, and tracking of physical objects.

As an example, the thermal sensor class in Fig. 3 is intended to include mechanical sensors whose *measurand* is thermal expansion/contraction and it includes the attribute *sample_interval* and provides two-signal processing services: a) *getExpansion()* which returns the expansion/contraction at regular sample intervals, and b) *detectAlarm(threshold)* which returns an alarm whenever the expansion/contraction crosses a specified threshold. Applications that leverage sensor ontologies will require a combination of sensor meta data, sensor data (sensor percepts), sensor services (capabilities), and the ability to reason about this information.

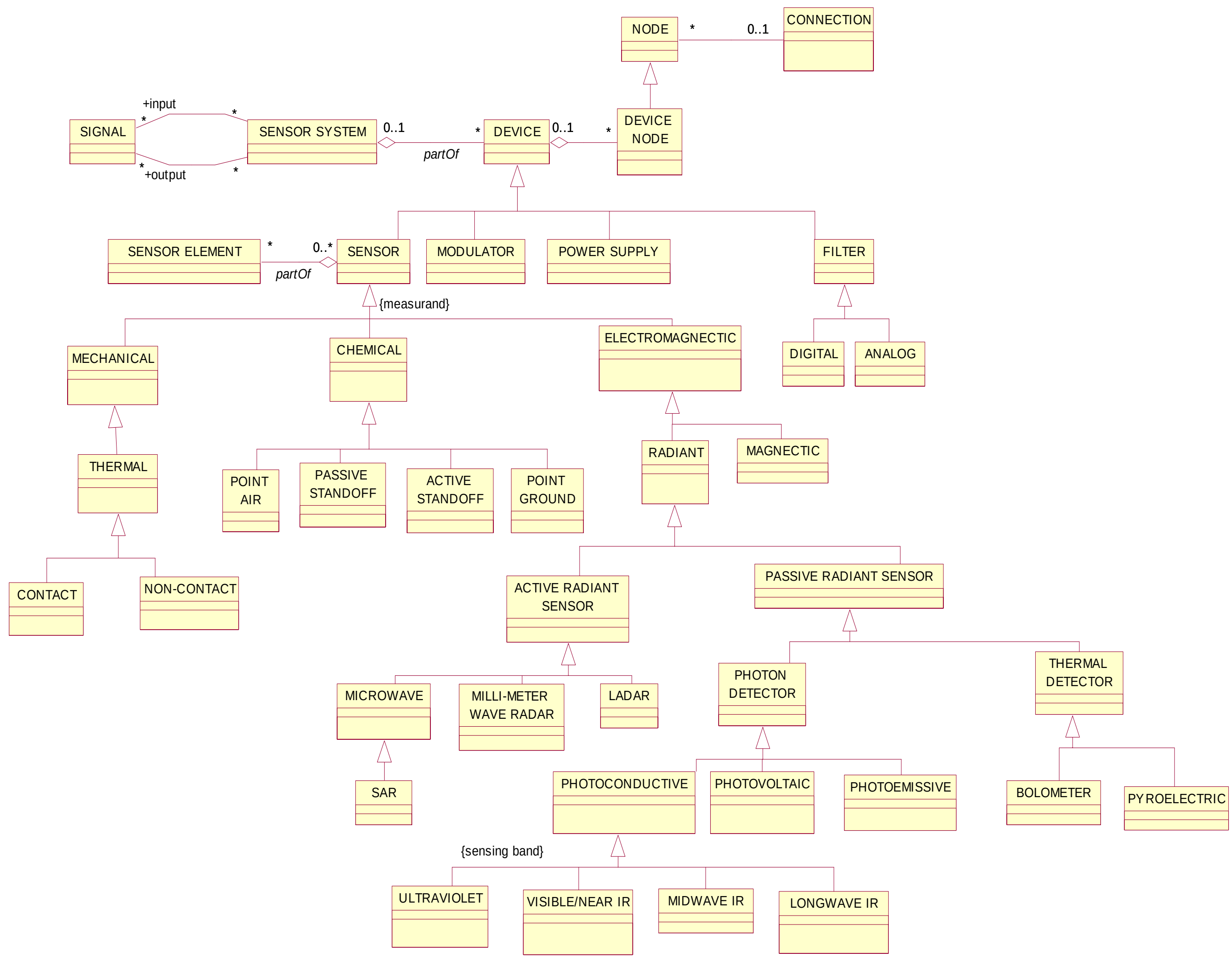

Fig. 2: Excerpt of a Possible Skeletal Upper-Sensor Ontology

C. Ontology-Based Sensor Simulations

Ontology-based sensor simulation applications will benefit from the query of sensor knowledge bases to provide realistic function and behavior of sensors and information required for fusion and interoperability in a simulation environment. Simulation environments will also require the following:

- definitions of all sensor services (*getExpansion* is an example sensor service)
- a programmer's interface (API) for sending messages to and receiving messages from the sensors
- a distributed query workload execution plan which determines the data (percepts) that should be extracted from the sensors' persistent data store

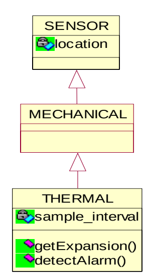

Fig. 3: Thermal Sensor

III. ONTOSENSOR: A PRACTICAL APPROACH

This section describes our practical approach to building a prototype sensor knowledge repository, referred to as OntoSensor, compatible with evolving Semantic Web infrastructure. OntoSensor is based on an ontology and is comprised of definitions of concepts and properties adopted in part from SensorML [8], extensions to the IEEE Suggested Upper Merged Ontology (Sumo) [9], references to ISO 19115 [10] and constructs from the Web Ontology Language. Initial sensor ontology instantiation and simple queries have been developed and tested using Protégé 2000 [11]. Although OntoSensor is in the early development stage, it presents a practical approach to building a sensor knowledge repository. It is proposed that OntoSensor will serve as a component in comprehensive applications that include more advanced inference mechanisms, which can be used for synergistic fusion of heterogeneous data.

SensorML is a specification of a generic data model in UML for capturing classes and associations that are common to all sensors. SensorML is part of an Open Geospatial Consortium (OGC) initiative to contribute to the development of a Sensor Web "through which applications and services will be able to access sensors of all types over the Web [8]." Instantiations of classes and associations provided by SensorML can be used to create specific sensor profiles, which facilitate the processing, geolocation and integration of observed data from a myriad of sensors. Profiles of individual sensors that use and/or extend SensorML concepts can be created and posted in a Web environment in which they can be tasked and queried by centralized monitoring and processing systems.

SensorML has been realized using syntax from the eXtensible Markup Language (XML) [12]. The use of XML syntax alone restricts the potential scope of interoperability and reuse of its instantiations. This is because of the lack of standard semantics of the XML syntactic constructs.

The lack of explicit semantics of the XML syntactic constructs necessitates the development of semantic mappings to translate between XML sensor profiles that are developed without a common reference. In general, given n profiles constructed with XML syntax $n(n-1)/2$ mappings have to be constructed to translate among the profiles.

XML syntactical constructs alone are unsuitable for the development of ontologies for the Semantic Web. Non-standard semantics of XML constructs alone preclude their sole use for formal definitions of sensor concepts. SensorML does not include formal definitions of the classes or relations it uses, that is, it provides no logical or axiomatic-grounded theory to account for its conceptualizations and therefore cannot be considered to be an ontology. However, SensorML does provide a generic data model for expressing knowledge and data about sensors and provides a good schema for developing a persistent data store for sensor metadata and sensed attribute values. Moreover, it provides a good organizational framework within which a sensor ontology can be defined.

OWL has been adopted by the World Wide Web Consortium (W3C) as a standard formalism for the Semantic Web. Each OWL construct has formal semantics. Part of the OntoSensor development effort includes mapping a subset of the SensorML concepts and relations to OWL; however several transformations and compromises are required in the implementation and are briefly discussed in subsequent sections. As a first step towards utilizing high-level sensor ontologies for data/knowledge fusion on the Semantic Web, OntoSensor is used to build a prototype sensor repository. This repository is an instantiation of the sensor ontology and is coupled with a simple inference module implemented in Prolog to build applications. This prototype is the predecessor of more advanced systems, which must include advanced inference capabilities that can realize the potential of using high-level sensor ontologies for sensor data fusion.

IV. ONTOSENSOR WORKFLOW

OntoSensor has been initially developed using the Protégé 2000 ontology editor. Following the creation of the OntoSensor knowledge base, a specific sensor is instantiated in Protégé 2000. Besides using Prolog for implementing OntoSensor applications, the Protégé 2000 querying tab is used to rapidly query and test instances of the knowledge base to obtain their properties and their capabilities.

A. From SensorML to OWL

The Protégé 2000 ontology editor, which provides an easy interface to define classes and properties, has been used to formally define the concepts and relations in OntoSensor. These definitions are then exported to OWL using the OWL plug-in feature of Protégé 2000 for posting to the Web. Although the initial objective was to faithfully replicate the concept hierarchy of SensorML in OWL, some implementation compromises and workarounds needed to be made during the creation of OntoSensor. These were necessary because of some limitations and constraints of the Protégé 2000 environment and due to the dependency of certain SensorML terms on concepts from the Geographic Markup Language (GML) [13]. Moreover, additional classes and associations will be required if richer models that extend Fig. 2 are to be supported.

B. Extending OntoSensor

Good ontological engineering includes attempting to leverage upper ontologies in which domain-independent knowledge is captured. Upper ontologies define general concepts and provide a common foundation for creating specialized domain-specific ontologies like OntoSensor. OntoSensor extends the IEEE SUMO upper-level ontology by making some OntoSensor classes extensions of classes defined in SUMO. Moreover, by using upper-level ontologies, a framework is deployed in which translation among different domain ontologies can be more readily accomplished. Furthermore, SensorML framework references some concepts that are defined in ISO 19115. The ISO 19115 standard defines schema required for geographic information and services. Both the SUMO and ISO 19115 have OWL implementations that are referenced by OntoSensor. Fig. 4 shows the structure of OntoSensor, which includes aspects of SensorML, extends IEEE SUMO and references ISO 19115.

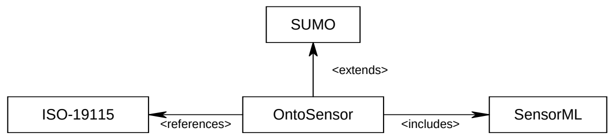

Fig. 4: OntoSensor Structure

OntoSensor extends IEEE SUMO by making the OntoSensor class *Event* a subclass of the SUMO class *SUMO:Process*, that is, an *OntoSensor:Event* is a kind of *SUMO:Process*. In OntoSensor, an *OntoSensor:Event* is a history of a sensor or the ontology itself kept in a series of events. For example, an instance of the event class can be the deployment of a sensor, or the deprecation of a class.

The OntoSensor class *OntologyEvent* was made a subclass of *SUMO:ContentDevelopment*. The OntoSensor class *Sensor* was made a subclass of *SUMO:MeasuringDevice*, and OntoSensor class *Platform* was made a subclass of *SUMO:TransportationDevice*.

One of the visions of SensorML is that the schema will be self describing; that is, meta-data about the schema will be contained within the schema. For example, the relations *documentConstrainedBy* and *documentedBy* on the class *Component* are two such relations that are self describing. These relations take objects that belong to the *Constraint* class as their respective ranges.

The implementation of OntoSensor also references the ISO 19115 constraints as presented in the SensorML schema. For example, the *legalUse* relation on the OntoSensor class *Constraint* takes the *ISO_19115:MD_LegalConstraints* as its range. Likewise the *securityLevel* relation on the OntoSensor class *Constraint* takes the *ISO_19115:MD_SecurityConstraints* as its range.

Additionally, meta-data about *Component*(s) can be captured with the *describedBy* relation, which accepts objects that belong to the OntoSensor class *ComponentDescription* as its range. The OntoSensor class *ComponentDescription* has five relations defined on it which all have the range of *ISO_19115:CI_ResponsibleParty*, *ISO_19115:CI_Citation*, or *ISO_19115:CI_OnlineResource*.

C. Dependencies on GML Concepts

Certain SensorML concepts pertaining to the location model of sensors, coordinate reference systems and transformation procedures are dependent upon concepts from the Geographic Markup Language (GML). These SensorML concepts are necessary for specifying the location of sensors and sensor observations and to perform spatial transformations to relate these to the location and reference systems of other sensors, platforms and central monitoring and processing systems.

This dependency upon GML can be captured in OntoSensor in one of two ways. The first way would require creating an OWL version of GML complete with formal definitions of GML concepts and referencing this new ontology in OntoSensor. The alternative would be to define these GML concepts in the OntoSensor namespace and eliminate the dependency upon GML altogether.

An OWL ontology with formal definitions of GML concepts is being developed by Defne et al. [14] using Protégé-2000. Referencing this ontology into the OntoSensor namespace is the preferred approach in the interests of knowledge reuse and modularity. In this way, the dependence of concepts in the OntoSensor ontology upon GML concepts is retained.

D. Querying the Knowledge Base

To initially demonstrate the utility of OntoSensor, five sensors were instantiated using the Protégé 2000. Based on these instances, the Protégé 2000 Queries plug in was used to query the knowledge base. Fig. 5 shows an excerpt of OntoSensor classes and associations which are currently implemented. Fig. 5 displays classes and relationships that capture sensors, the phenomenon they measure, and their capabilities. Redefinition of some of the classes in Fig. 5 may be necessary and is the topic of ongoing study. For example, many sensor capabilities may be better represented by pushing them down to specific sensor types rather than linking them directly to a generic sensor class.

In Fig. 5, the *hasCapabilities* relation is defined on the domain of *Sensor* and the range of *CapabilitiesDescription*; this relation links a sensor and its corresponding capabilities. In SensorML schema, the attribute *GenericProperty* has been reified to a class to capture added information of a generic property. In OntoSensor, this class has been extended with 13 subclasses that specializes the property. Examples of the specialization of *GenericProperty* include *SearchArea*, *RangeData*, etc.

Fig. 6 shows a query that returns a sensor (mmwr1) that has a capability of large search area. Two joins are performed to retrieve the solution. First a join is performed on the class *GenericProperty* and *CapabilitiesDescription*, and then another join is performed on *CapabilitiesDescription* and *Sensor*.

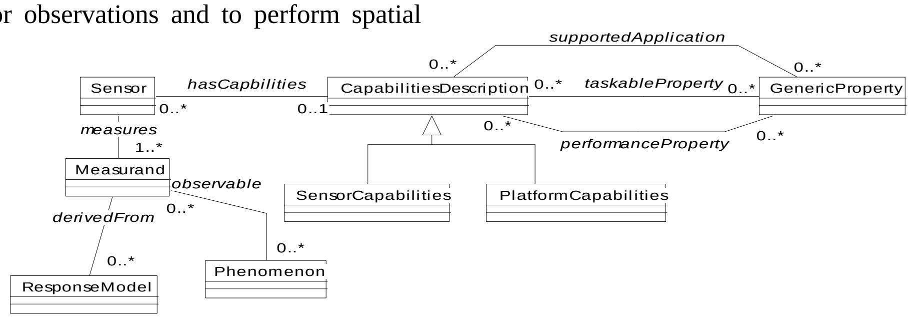

Fig. 5: Snapshot of the Sensor Package

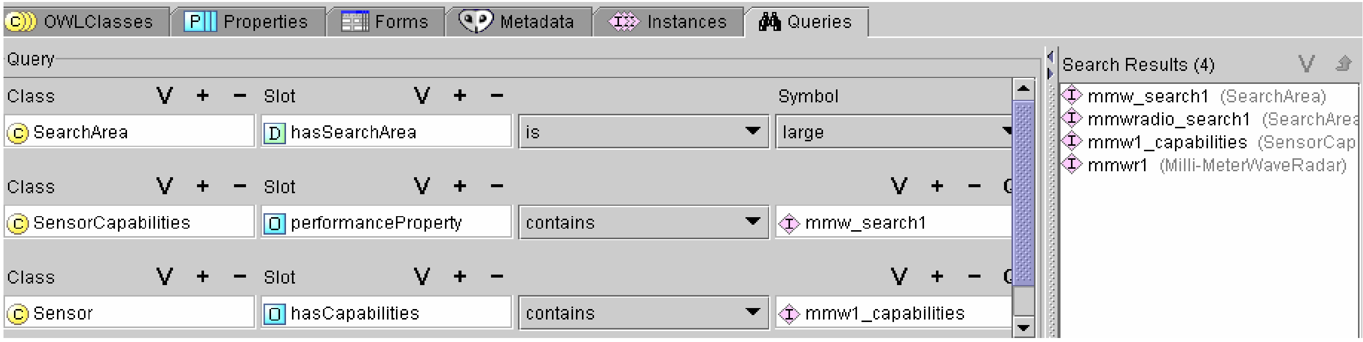

Fig. 6: Querying OntoSensor using Protégé 2000 Plug In

V. CONCLUSIONS

An ontology containing the definitions of high-level concepts pertaining to sensors can be used as background knowledge for the integration of data from heterogeneous sensors. Developing such an ontology, which is compatible with the Semantic Web, will leverage this evolving infrastructure for network-centric sensor data fusion. On the Semantic Web, data can be queried readily from profiles of sensors and groups of sensors can be tasked to work in coordination in a comprehensive environment.

We have described our approach to developing an ontology-based, prototype sensor repository referred to as OntoSensor. OntoSensor provides formal definitions of the concepts and relations influenced from SensorML. OntoSensor extends concepts from the IEEE SUMO ontology, and references terms from ISO 19115.

We are currently attempting to provide a pragmatic repository of representative imaging sensors, with supporting rationale and using currently available tools, which we propose will lead toward the deployment of sensor ontologies in a variety of application domains. Future directions include the investigation of what aspects and detail of traditional, physics-based sensor models are feasible for representation in OntoSensor. And as such, is the current model and supporting infrastructure (OWL, Protégé 2000, etc.) adequate for capturing the additional detail that may be needed for advanced applications? Also, how non-ontology based implementation standards being developed by organizations such as OGC and ISO can be best leveraged to develop deep ontologies is of continued interest.

VI. ACKNOWLEDGEMENT

Funding for this work was provided in part by U.S. Government Contract DAAB07-01-D-G601 and by ERC Inc., under purchase orders G6002681 and ET040991, respectively. This paper does not necessarily represent the position of U.S. Government agencies or ERC Incorporated.

VII. REFERENCES

[1] NAC (National Research Council), *Expanding the vision of sensor materials*, Washington, D.C.: National Academy Press, 2000.

[2] G. Jiang, W. Chung, and G. Cybenko, "Semantic Agent Technologies for Tactical Sensor Networks," *2003 SPIE Conference on AeroSense*.

[3] T. Berners-Lee, J. Hendler and O. Lassila, "The Semantic Web: A new form of Web Content that is meaningful to Computers will unleash a revolution of new possibilities," *Scientific American*, May 2001, pp. 34-43.

[4] M. Ceruti, "Ontology for Level-One Sensor Fusion and Knowledge Discovery," *8th European Conference on Principles and Practice of Knowledge Discovery in Databases*, Pisa, Italy, 2004.

[5] R. Jurdak, C. Lopes and P. Baldi, "A Framework for Modeling Sensor Networks," *OOPSLA Workshop on Building Software for Pervasive Computing*, 2004.

[6] M. Genesereth and N. Nilsson, *Logical Foundations of Artificial Intelligence*, Palo Alto, CA: Morgan Kaufmann Publishers, 1987.

[7] M. Smith, C. Welty and D. McGuinness, "OWL Web Ontology Language Guide," W3C Recommendation, February 10, 2004. Available at: http://www.w3.org/TR/owl-guide/

[8] M. Botts et al., "Sensor Model Language (SensorML) for In-Situ and Remote Sensors," Sep. 2004. Available at: http://vast.nsstc.uah.edu/SensorML/Sensor_Model_Language_SensorML_for_Insitu_and_Remote_Sensors.doc.

[9] I. Niles and A. Pease, "Origins of the Standard Upper Merged Ontology: A Proposal for the IEEE Standard Upper Ontology," In *Working Notes of the IJCAI-2001 Workshop on the IEEE Standard Upper Ontology*, Seattle, WA, 2001.

[10] International Organization for Standardization, Text for FDIS 19115 Geographic information – Metadata. Final Draft Version. ISO/TC-211, 2003. Available at: http://www.ncits.org/ref-docs/FDIS_19115.pdf

[11] Stanford Medical Informatics, "The Protégé Ontology Editor and Knowledge Acquisition System," October 2004. Available at: http://protege.stanford.edu/

[12] World Wide Web Consortium, "Extensible Markup Language (XML)," August 2004. Available at: http://www.w3.org/XML/

[13] Open Geospatial Consortium, "GML – A Markup Language for Geography," June 2004. Available at: http://opengis.net/gml/

[14] Z. Defne, A. Islam and M. Piasecki, "*Ontology for Geography Markup Language (GML3.0) of Open GIS Consortium (OGC),*" September 2004. Available at: http://loki.cae.drexel.edu/~wbs/ontology/2004/09/ogc-gml.owl