

Building a Sensor Ontology: A Practical Approach Leveraging ISO and OGC Models

David J. Russomanno, Cartik R. Kothari and Omoju A. Thomas
Department of Electrical and Computer Engineering
The University of Memphis
Memphis, TN 38152 USA

Abstract - This paper describes the approach to building OntoSensor: a prototype sensor knowledge repository compatible with evolving Semantic Web infrastructure. OntoSensor includes definitions of concepts and properties adopted in part from SensorML, extensions to IEEE SUMO and references to ISO 19115. Simple queries have been developed and tested using Protégé 2000 and Prolog. Although OntoSensor is in the early development stage, it presents a practical approach to building a sensor knowledge repository. It is proposed that OntoSensor may serve as a component in comprehensive applications that include more advanced inference mechanisms. Such comprehensive applications will be used for synergistic fusion of heterogeneous data in a network-centric environment.

Keywords: Data Fusion, Semantic Web, Sensor Ontology

1.0 Introduction

The assessment of situations and strategies in complex environments requires the fusion of information from heterogeneous data sources including sensors [1]. Synergistic fusion of data from multiple sensors ranging in complexity from simple acoustic sensors to sophisticated imaging equipment such as Forward Looking Infrared (FLIR) sensors will lead to the extraction of knowledge that cannot be perceived or inferred using individual sensors alone. For example, information from diverse sources such as sensors on an Unmanned Aerial Vehicle (UAV), reconnaissance reports and satellite imagery can be integrated to obtain high-level knowledge of objects in an area under surveillance including their spatial and temporal interrelationships; and to generate predictions of their intentions, positions and alignments in future states.

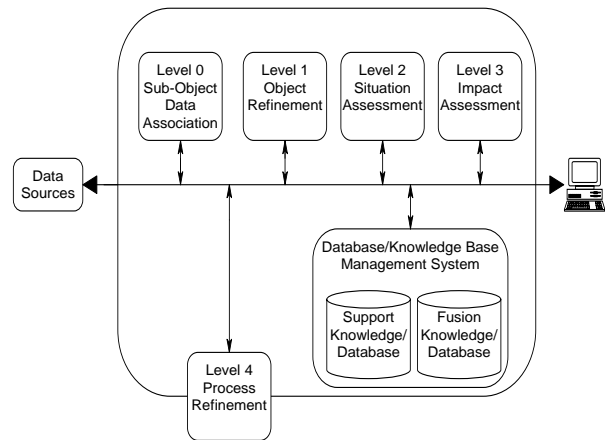


Figure 1: JDL data fusion model [3]

In military applications, the fusion of information from sensors and other heterogeneous data sources may enable the assessment of enemy tactics and strategies and the development of counter strategies and tactics. The Data Fusion Group of the Joint Directors of Laboratories (JDL) Technical Panel for Command & Control, Communications, Computers and Intelligence (C4I) developed a high-level functional model of the data fusion process [2] that was later extended by Llinas et al. [3]. This model defines common nomenclature used in the data fusion community including the notion of fusion processing levels. It is proposed that OntoSensor would serve as a component of the support knowledge base in JDL-inspired applications.

There are several impediments to widespread, heterogeneous data and knowledge fusion. The *Committee on New Sensor Technologies: Materials and Applications* [4] has implied that the potential of advanced sensor technology has been limited by the lack of a well-developed and

common language for expressing sensor definitions, attributes, classifications (including taxonomies), as well as descriptions of sensor needs and performance. Typically, technical specifications of sensors differ not only in terms of parameters and their values, but also in the terminology that is used to define them, which limits their use in fusion processes. To partially address this issue, comprehensive sensor ontologies are needed to establish a widely accepted terminology of sensors, their properties, capabilities, and services. OntoSensor should be viewed in this context.

This paper describes a practical approach to building a sensor ontology using the SensorML specification [5], IEEE Suggested Upper Merged Ontology (SUMO) [6], International Organization for Standardization (ISO) 19115 standards [7], and constructs of the Web Ontology Language (OWL) [8].

The remainder of this paper is organized as follows. Section 2 discusses SensorML, its role as a building block for developing OntoSensor, and the reasons that SensorML alone is insufficient as an ontology. Section 3 presents an overview of the OntoSensor workflow, including sensor instantiation and the design compromises that were necessary for OntoSensor's implementation. Section 4 illustrates some sample queries of OntoSensor and Section 5 provides conclusions and future directions.

2.0 SensorML: A Building Block

SensorML is specified as 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 [5]." 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 monitoring and processing systems.

SensorML is developed as a specification for efficient implementation by vendors that desire to implement OGC compliant sensor systems; therefore, SensorML strives to specify as few class and relation definitions as possible. However, ontological engineering focuses on rich semantic data and knowledge models; therefore, re-conceptualization, redefinition, or extension of some of the classes in SensorML is necessary.

SensorML has been realized using syntax from the eXtensible Markup Language (XML) [9]. The use of XML syntax alone to define classes restricts the potential scope of interoperability and reuse of its sensor profile instantiations. This is because of the lack of standard semantics of the XML syntactic constructs.

In general, given n sensor profiles constructed with XML syntax, $n*(n-1)/2$ mappings have to be constructed to translate among the profiles. Further, the lack of semantics of XML constructs alone precludes their use for formal definitions of sensor concepts in an ontology. According to Gruber [10], an ontology is a set of formally defined concepts and relations that are relevant to a knowledge domain. 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 [11]. Each OWL construct has formal semantics. Part of the OntoSensor development effort includes mapping a subset of the SensorML concepts and relations to OWL and adding additional concepts. Several transformations and compromises were required in the implementation and some are highlighted in Section 3.

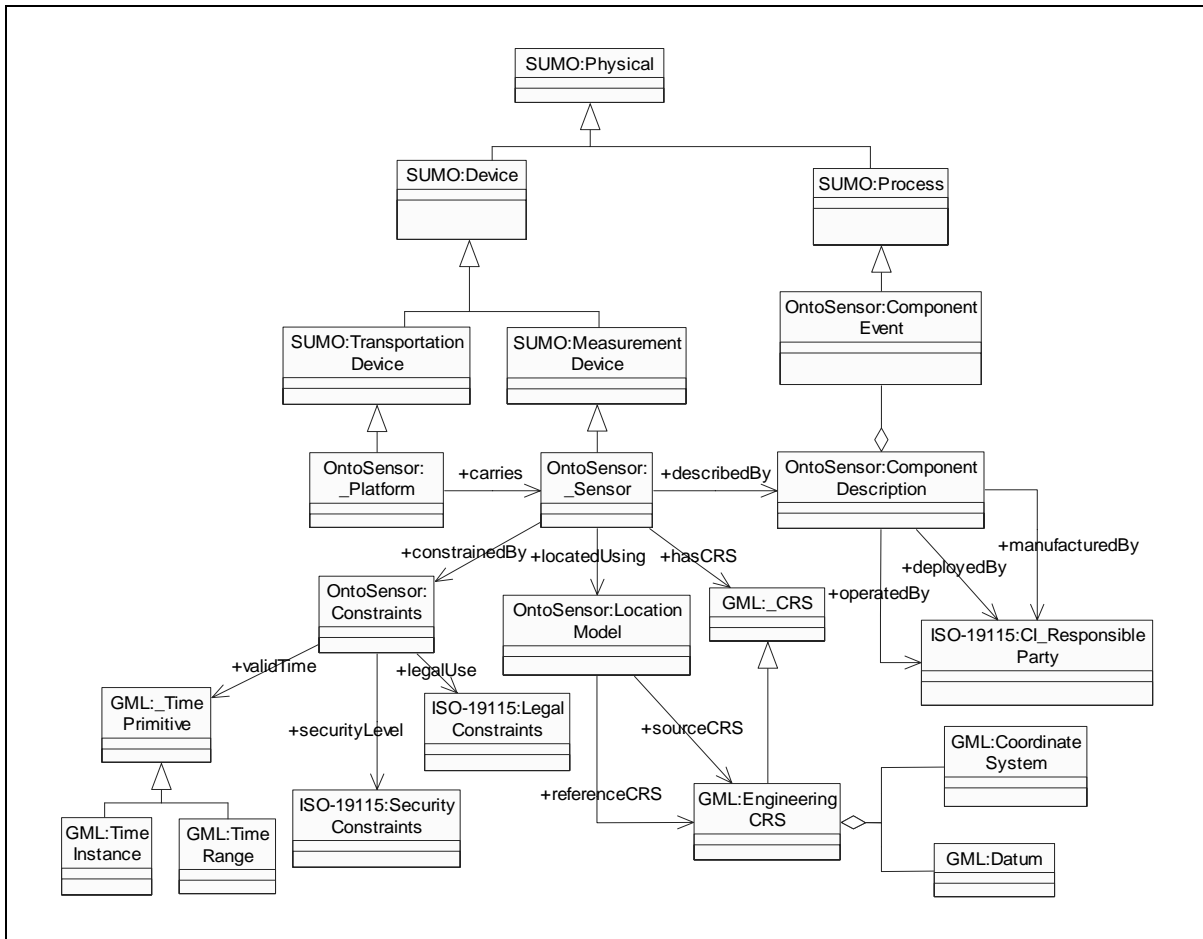


Figure 2: OntoSensor extends IEEE Sumo concepts (abridged)

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 in knowledge and data fusion processes.

3.0 OntoSensor Workflow

OntoSensor has been initially developed using the Protégé 2000 ontology editor [12]. 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.

3.1 From SensorML to OWL

After formally defining OntoSensor concepts and relations using the Protégé 2000 ontology editor, these definitions are then exported to OWL using the OWL plug-in feature of Protégé 2000 for posting to the Web. Although the 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].

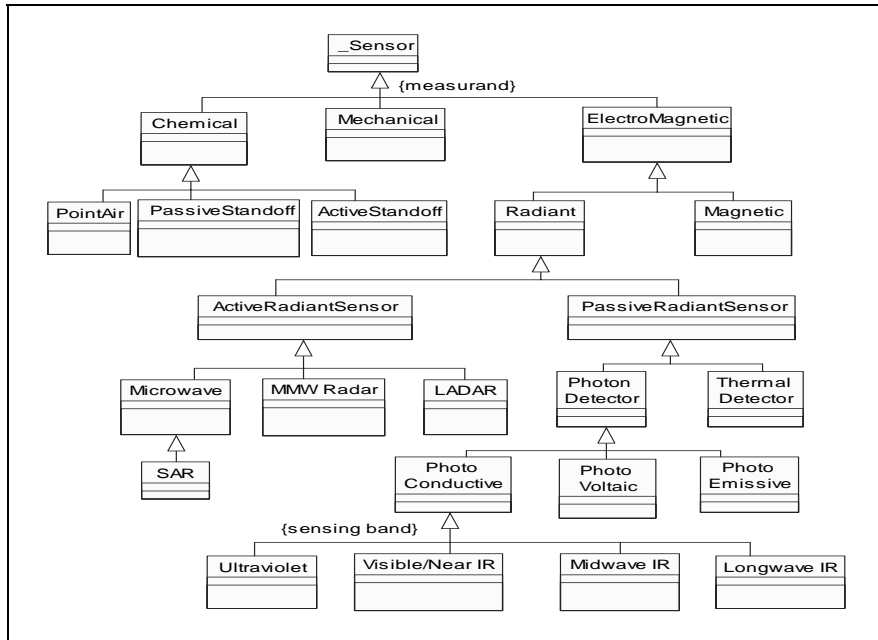


Figure 3: Excerpt of sensor hierarchy within OntoSensor

3.2 Extending OntoSensor

Good ontological engineering includes attempts 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, the SensorML specification 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. 2 shows an aspect of the concept hierarchy of the OntoSensor ontology which includes formal definitions of SensorML concepts, extends IEEE SUMO and uses concepts from ISO 19115 in some of its relations. In Fig. 2, the *_Sensor* and *_Platform* concepts of OntoSensor extend the *MeasurementDevice* and

TransportationDevice concepts of the IEEE SUMO ontology, respectively. The *Event* concept of OntoSensor extends the *Process* concept of SUMO. 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 relation *constrainedBy* on the class *_Sensor* is a self-describing relation. This relation takes objects that belong to the *Constraints* class as its range.

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 *Constraints* takes the *ISO-19115:LegalConstraints* as its range. Likewise the *securityLevel* relation on the OntoSensor class *Constraints* takes the *ISO-19115:SecurityConstraints* as its range.

Additionally, meta-data about *Sensor(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* serves as the domain in three relations, all of which have as their range the class *ISO-19115:CI_ResponsibleParty*.

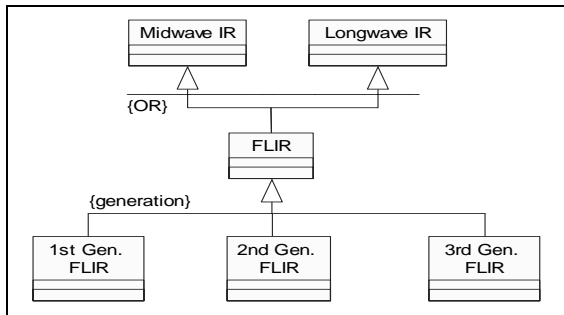


Figure 4: FLIR extension to OntoSensor

An excerpt of the hierarchy of sensor types within OntoSensor is shown in Fig. 3. The root concept in this hierarchy is the *_Sensor* concept, which extends the *MeasurementDevice* concept from SUMO (ref. Fig. 2). Specific sensor instances can extend concepts from the OntoSensor hierarchy. For example, a FLIR sensor can sense emissions in the long wave and/or medium wave infrared range. Specifically, 1st and 2nd generation FLIR sensors can detect emissions in either of these ranges while 3rd generation FLIR sensors can detect emissions across both ranges. This is shown by the class diagram in Fig. 4.

As shown here, OntoSensor can be viewed as a middle-level ontology that extends the concepts from a high-level ontology (SUMO) and whose concepts can be used by more specialized ontologies that model specific domain sensors such as FLIR.

3.3 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 into OntoSensor. The alternative would be to define

the requisite subset of GML concepts entirely within the OntoSensor namespace and eliminate the dependency upon GML altogether.

An OWL ontology with formal definitions of GML concepts is being created 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.

As shown in Fig. 2, the dependence on GML is captured by the *hasCRS* relation between the *_Sensor* concept of OntoSensor and the *_CRS* concept from GML. Next, the *_Sensor* concept is associated with a *LocationModel* concept that uses the *EngineeringCRS* concept from GML for its reference and source CRS. Note that the *EngineeringCRS* concept is an aggregate of the *CoordinateSystem* and *Datum* concepts from GML.

3.4 Example Sensor Instance

Instantiations of some OntoSensor classes were created using the Protégé 2000 instances tab plug-in.

```
<FLIR rdf:ID="FLIR_001">
<hasCapabilities>
<SensorCapabilities rdf:ID="FLIR_001_capabilities">
<supportedApplication
rdf:resource="#Fineresolutionimagery"/>
<supportedApplication
rdf:resource="#Daynightoperation"/>
<supportedApplication
rdf:resource="#Covert"/>
<performanceProperty>
<GenericProperty rdf:ID="OntoSensor_Individual_337">
<Attr_Name rdf:datatype=
"http://www.w3.org/2001/XMLSchema#string">
Overscan ratio</Attr_Name>
<Attr_Value rdf:datatype=
"http://www.w3.org/2001/XMLSchema#string">
1</Attr_Value>
</GenericProperty>
</performanceProperty>
<performanceProperty>
<GenericProperty rdf:ID="OntoSensor_Individual_339">
<Attr_Name rdf:datatype=
"http://www.w3.org/2001/XMLSchema#string">
Number of detectors</Attr_Name>
<Attr_Value rdf:datatype=
"http://www.w3.org/2001/XMLSchema#string">
180</Attr_Value>
</GenericProperty>
</performanceProperty>
</SensorCapabilities>
</hasCapabilities>
</FLIR>
```

Figure 5: Excerpt of OntoSensor instance

Fig. 5 displays an excerpt of the OWL file generated as the output of the instantiation. The excerpt shows the OWL constructs that capture the following knowledge: The resource *FLIR_001* is an instance of the OntoSensor FLIR class. This instance of a FLIR sensor is appropriate for obtaining fine resolution imagery, has day and night operability, and can be used for projects that require some concealment; hence the sensor also has a covert capability. In addition, *FLIR_001* also has an overscan ratio of 1 and is equipped with 180 detectors.

4.0 Querying the knowledge base

Fig. 6 shows an excerpt of a class diagram used in OntoSensor. A sensor's capability has been captured through the class *CapabilitiesDescription*. This class in turn is linked to the *GenericProperty* class through two associations. A specific sensor's sensitivity and resolution can be derived through the *performanceProperty* association. For instance, noise equivalent temperature difference (NETD), which is one gross measure of sensitivity, can be determined by obtaining the values of certain sensor parameters like focal length, horizontal and vertical field of view, frame rate, overscan ratio and so on [15]. Through the *supportedApplications* association, queries can determine the kind of sensor to deploy based on some application criteria like day/night operation capability, foliage penetration, all weather capability and so on.

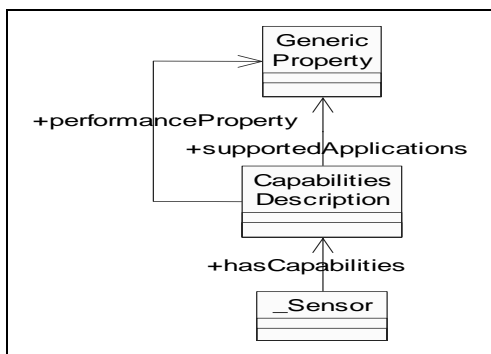


Figure 6: Excerpt of OntoSensor model

Initially it was proposed that the creation and querying of OntoSensor be done strictly in the Protégé 2000 environment. However, as the

project progressed, it became apparent that the Protégé 2000 querying interface was not well suited for handling complex queries that needed to be issued to the OntoSensor knowledge base. SWI-Prolog [16] was adopted because of its built-in inference mechanisms, its support of Semantic Web technologies, and its rapid prototyping capability.

```

1 ?-sensor_capability('FLIR_001',
    ListOfCapabilities).
ListOfCapabilities =
['Covert',
'Day/Night Operation',
'Fine Resolution Imagery']

2 ?-sensor_capability(SensorInstance,
    'Fine Resolution Imagery').
SensorInstance = LASER_001 ;
SensorInstance = FLIR_001

3 ?-sensor_parameters('FLIR_001',
    ListOfParameters).
ListOfParameters = [
'Entrance aperture diameter'='20 cm',
'Focal Length'='35 cm',
'Frame rate'='30 Hz',
'Horizontal DAS'='0.2 mrad',
'Horizontal FOV'='4 deg',
'Interlace'='2:1',
'Number of detectors'='180',
'Optics Transmission'='0.70',
'Overscan ratio'='1',
'Scan efficiency'='0.75',
'Vertical DAS'='0.2 mrad',
'Vertical FOV'='3 deg']
  
```

Figure 7: Example query result

Fig. 7 shows a representative query of the OntoSensor knowledge base using SWI-Prolog's command line. Line one is a query that takes an instance of a FLIR sensor and determines its capabilities. The second query finds all sensors in the knowledge base that can capture fine resolution images. Line three shows a query that retrieves the parameters for a FLIR instance.

5.0 Conclusions

In this paper, the development of a prototype sensor repository that uses the OntoSensor ontology to instantiate its knowledge base has been described. OntoSensor provides formal definitions of the concepts and relations influenced from SensorML. The concepts from OntoSensor extend concepts from the IEEE SUMO ontology, and also reference terms from the ISO 19115 standard, all of which are represented in OWL.

Although we make no claims that an orthogonal, complete or universally acceptable

sensor ontology is feasible, we are 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 augmenting OntoSensor with service specifications, procedural attachment, and 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, OWL-S, Protégé 2000, etc.) adequate for capturing the additional detail and capabilities that may be needed for advanced applications?

6.0 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 Inc.

7.0 References

- [1] D. Hall and J. Llinas, "Multisensor Data Fusion," In D. Hall and J. Llinas (Eds.), *Handbook of Multisensor Data Fusion*, CRC Press, 2001.
- [2] F. White, "A Model for Data Fusion," In *Proceedings of the 1st National Symposium on Data Fusion*, 1988.
- [3] J. Llinas et al., "Revisions and Extensions to the JDL Data Fusion Model II," In *Proceedings of the 7th International Conference on Information Fusion*, Stockholm, Sweden, 2004, pp. 1218–1230.
- [4] NRC (National Research Council), *Expanding the vision of sensor materials*, Washington, D.C.: National Academy Press, 2000.
- [5] M. Botts et al., "Sensor Model Language (SensorML) for In-Situ and Remote Sensors," November 2004. Available at: http://vast.nsstc.uah.edu/SensorML/SensorML_04-019_1.0_beta.pdf
- [6] 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.
- [7] 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
- [8] P. Patel-Schneider, P. Hayes and I. Horrocks, "OWL Web Ontology Language Semantics and Abstract Syntax," W3C Recommendation, February 2004. Available at: <http://www.w3.org/TR/owl-semantics/>
- [9] World Wide Web Consortium, "Extensible Markup Language (XML)," August 2004. Available at: <http://www.w3.org/XML/>
- [10] T. Gruber, "A Translation Approach to Portable Ontologies," *Knowledge Acquisition*, 5(2): 199 – 220, 1993.
- [11] T. Berners-Lee, O. Lassila and J. Hendler, "The Semantic Web: A New Form of Web Content that is Meaningful to Computers will unleash a Revolution of New Possibilities," *Scientific American*, May, pp. 34–43, 2001.
- [12] Stanford Medical Informatics, "The Protégé Ontology Editor and Knowledge Acquisition System," October 2004. Available at: <http://protege.stanford.edu/>
- [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)," 2004. Available at: <http://loki.cae.drexel.edu/~wbs/ontology/2004/09/ogc-gml.owl>
- [15] R. Driggers, P. Cox and T. Edwards, *Introduction to Infrared and Electro-Optical Systems*, Boston, MA: Artech House, 1999, pp. 281–331.
- [16] J. Wielemaker, "SWI-Prolog Reference Manual," Online Resource Available at: <http://www.swi-prolog.org/>, 2000.