ABSTRACT

The development of an interdisciplinary virtual observatory (the Virtual Solar-Terrestrial Observatory; VSTO) as a scalable environment for searching, integrating, and analyzing databases distributed over the Internet requires a higher level of semantic interoperability than heretofore required by most (if not all) distributed data systems or discipline-specific virtual observatories. The formalization of semantics using ontologies and their encodings for the Internet (e.g., using OWL - the Web Ontology Language), as well as the use of accompanying tools, such as those supporting reasoning, analysis, and explanation, open up both a substantial leap in options for interoperability and in the need for formal development principles to guide ontology development and use within modern, multi-tiered network data environments. In this presentation, we outline the formal methodologies we are utilizing in the VSTO project, the currently developed use-cases, ontologies and their relation to existing ontologies (such as SWEET).

Domain Application

VSTO comprises a framework which provides virtual access to specific solar, solar-terrestrial and space-physics data, model, tool and material archives containing items from a variety of space- and ground-based instruments and experiments, as well as individual and community modeling and software efforts bridging research and educational use.

Datasets alone are not sufficient to build a virtual observatory. The VSTO addresses the interface problem to bring data to the users’ tools, and to the tools within the VSTO, effectively and with scalability. VSTO leverages the development of schema (e.g., CEDAR, MLSO, Earth System Grid, SPML), that adequately describe the syntax (name of a variable, its type, dimensions, etc. or the procedure name and argument list, etc.) and semantics (what the variable physically is, its units, etc. or what the procedure does and returns, etc.) of the datasets and tools.

Ontology Development

Use ontologies to power enhanced interoperable search tools
Build/evolve: using use-case applications as a foundation to define the ontology helps provide a comprehensive framework to build on.
Extract vocabulary: from use-case, with help of domain experts
Identify classes, properties, individuals: with help from knowledge representation experts (in collaboration with domain experts)

Build small ontology: vsto.owl, cedar.owl, MLSO.owl

Instances: enhanced with explicit metadata

Outcomes

Successful development of core ontology from two use-cases from two different disciplines exploiting existing standard ontologies (SWEET)
Successful automatic generation of interface elements for use-cases
Successful integration with existing metadata records (SQL databases), data retrieval service (OPeNDAP) and plotting procedures (ION and IDL) to implement two use-cases
Successful use of reasoning engine to infer plotting types for selected parameters
Path forward includes expanding functionality of use-cases supporting flexible, context knowledgeable interfaces along with increased breadth and depth to cover wider domain coverage