

Enhancing Grid Usage through Semantic Metadata

Alexandre César Tavares Vidal
Universidade Federal do Maranhão

Sergio Takeo Kofuji
University of São Paulo, Brazil

June 2007



- Introduction
- The knowledge Base
- The Semantic Grid System
- Conclusion



The Problem of Grid Resource Sharing

- Increasing amount of installed applications in Grid environments causes:
 - Increasing difficulty of finding a desired application, but also...
 - Increasing chances of software pieces interchange in response to a user application requirement;
- How to explore such reuse opportunities?



The Semantic Approach

- Definition of Grid Ontologies
 - To improve the reuse, sharing, and integration of software and computational resources on the grid;
 - Inference of information from axioms defined in ontologies.



The Semantic Approach

- Definition of Grid Ontologies
 - Using the W3C-recommended OWL-DL description language;
 - Using the Pellet reasoner for inference tasks;
 - Protégé-OWL to ease ontology description;
 - The opportunistic grid middleware, InteGrade, is the base for a prototype implementation.



Defining the ontologies

- Set of related ontologies, connected between them through the OWL *import* mechanism;
 - *upper-level ontologies*
 - higher level of generality;
 - *ancillary ontologies*
 - auxiliary domain specific concepts;
 - *concrete grid ontologies*
 - grid environment specific concepts



Upper level ontology – Grid Base Ontology

GridBaseOntology Protégé 3.2 beta (file:/home/vidal/public_html/GBO/GridBaseOntology.pprj, OWL / RDF Files)

File Edit Project OWL Code Tools Window Help

Metadata (Ontology1173179886.owl) OWLClasses Properties Individuals Forms

SUBCLASS EXPLORER

For Project: GridBaseOntology

Asserted Hierarchy

- owl:Thing
 - po:ArchitectureType
 - po:OperatingSystem
 - po:Processor
 - GridResourceConcepts
 - Cluster
 - Computer
 - DiskSpace
 - MemorySpace
 - Platform
 - GridSoftwareConcepts
 - Algorithm
 - Approach
 - Domain
 - Problem
 - SoftwareArtifact
 - Application
 - Configuration
 - Executable

CLASS EDITOR

For Class: Executable (instance of owl:Class) Inferred View

Annotations

Property	Value	Lang
rdfs:comme...		

Asserted Conditions

NECESSARY & SUFFICIENT

- implements some Application
- runsOn some Platform

NECESSARY

- SoftwareArtifact

Disjoints

Logic View Properties View



Ancillary ontology – Platform Ontology

The screenshot displays the Protégé 3.2 beta interface for the PlatformOntology project. The main window is divided into two panes, both titled 'SUBCLASS EXPLORER'.

The left pane, 'Asserted Hierarchy', shows a tree structure of classes. The 'Processor' class is expanded, showing its subclasses: 'AMD' (with 'AMD64' selected) and 'Intel' (with 'Celeron', 'Itanium', 'Pentium', and 'x86' listed). Other classes include 'OSArchitecture', 'OSProduct', 'BSD', 'MacOs', 'Unix', and 'Windows'.

The right pane, 'Inferred Hierarchy', shows the same tree structure but with additional inferred subclasses. Under 'Processor', it includes 'Processor32' (with 'x86' listed), 'Processor32_64' (with 'AMD64' listed), and 'Processor64' (with 'Itanium' listed). Under 'Unix', it includes 'Linux' (with 'Debian' and 'Ubuntu' listed, where 'Ubuntu' is further expanded to show 'Ubuntu6_10').

The interface includes a menu bar (File, Edit, Project, OWL, Code, Tools, Window, Algernon, Help), a toolbar with various icons, and a status bar at the bottom with additional icons.



Concrete ontology – Software Management Ontology

SoftwareManagementOntology Protégé 3.2 beta (file:/home/vidal/public_html/SoftwareManagementOntology/SoftwareM...)

File Edit Project OWL Code Tools Window Help

Metadata (protege) OWLClasses Properties Individuals Forms

SUBCLASS EXPLORER
For Project: SoftwareManagementOntolo...
Asserted Hierarchy

- owl:Thing
 - rdf:Property
 - p1:GridSoftwareConcepts
 - p1:SoftwareArtifact
 - p1:Domain
 - p1:Problem
 - MathematicProblem
 - BiologyProblem
 - DataMiningProblem
 - p1:Algorithm
 - ClassificationAlgorithm
 - CHAID_Algorithm
 - ClusteringAlgorithm
 - DataMiningAlgorithm
 - HierarchicalClusteringAlgorithm
 - LinkAnalysisAlgorithm
 - ProbabilisticClusteringAlgorithm
 - MCLUST_Algorithm
 - SLIQ_Algorithm

SUBCLASS EXPLORER
For Project: SoftwareManagementOntology
Inferred Hierarchy

- p1:GridResourceConcepts
- p1:GridSoftwareConcepts
 - p1:Algorithm
 - DataMiningAlgorithm
 - ClassificationAlgorithm
 - CHAID_Algorithm
 - SLIQ_Algorithm
 - ClusteringAlgorithm
 - HierarchicalClusteringAlgorithm
 - ProbabilisticClusteringAlgorithm
 - MCLUST_Algorithm
 - LinkAnalysisAlgorithm
 - p1:Approach
 - p1:Domain
 - p1:Problem
 - p1:SoftwareArtifact
 - p1:Application
 - DataMiningApplication
 - ClassificationApplication
 - ClusteringApplication

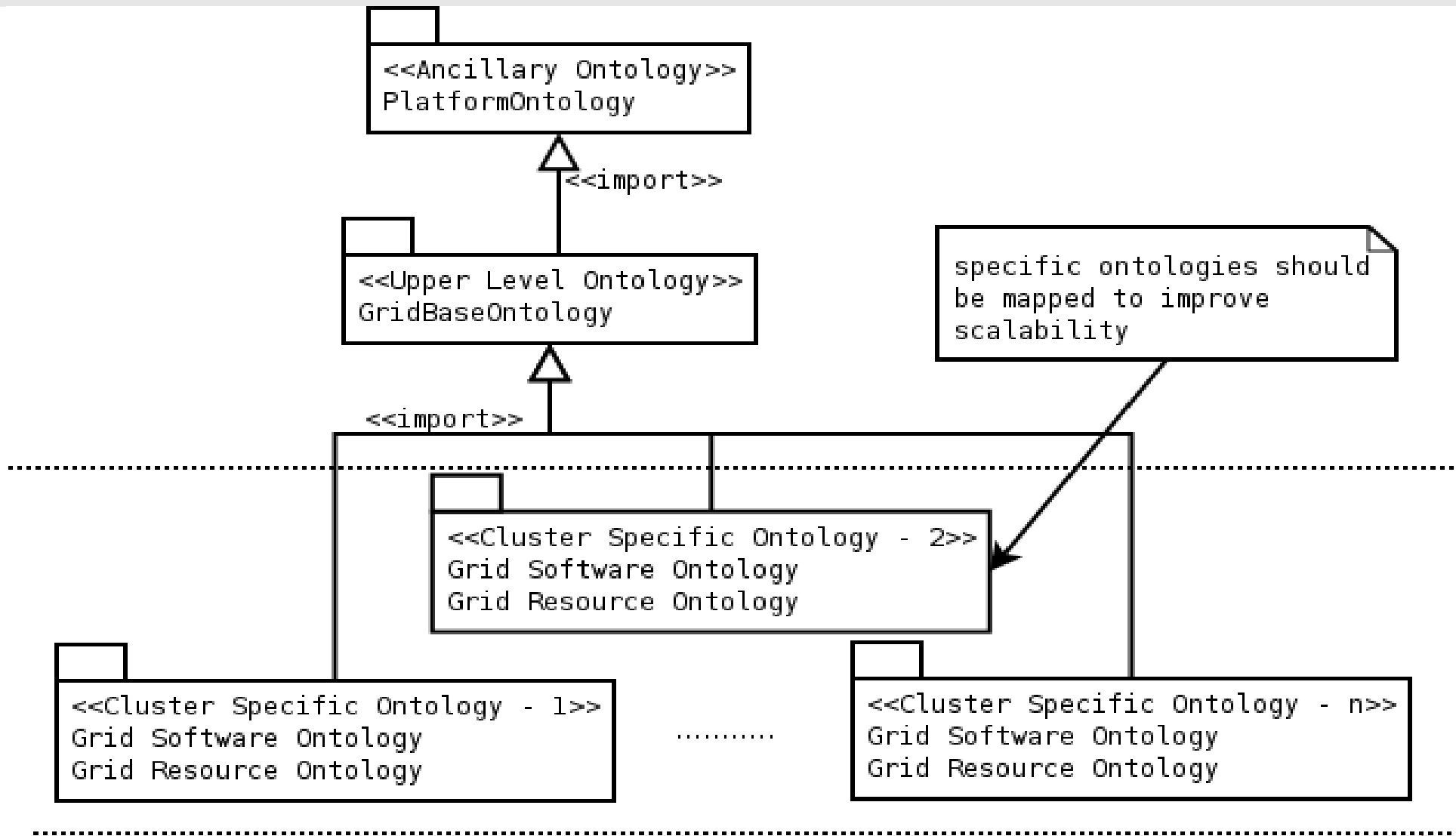


Exploring the ontologies

- Fundamental taxonomy encompassing the main concepts related to grid systems;
- Inference knowledge from previously defined axioms and incomplete information, e.g.:
 - subsumption inference in advance, in conformance with an intelligent policy;
 - query languages and mechanisms;
 - SPARQL;
 - OWL-QL



Extending the ontologies



Grid Ontologies Applicability

- New inferred subsumed hierarchies to obtain new inferred knowledge;
- Ontology-based applications can be built around the KB to cover different domain problems;
- More efficient query results from inferred class hierarchies;
- improve application and grid resource matching.

