MGC'06 November 27-December 1, 2006 Melbourne, Australia

Defining and Exploring a Grid System Ontology

Alexandre C.T. Vidal **{ vidal@ime.usp.br }**
José de R. Braga P. Júnior { jrbraga@ime.usp.br }
Fabio Kon { kon@ime.usp.br }
Dept. of Computer Science

Sérgio T. Kofuji { kofuji@lsi.usp.br }
Dept. of Electronic Systems Engineering
University of São Paulo, Brazil

Site: http://www.integrade.org.br/portal/GridOntology

Objective and Motivation

Objective

This work introduces a generic and extensible grid system ontology.

Motivation

Grid computing involves coordinated sharing of resources for the resolution of computational problems in dynamic and heterogeneous environments.

Ontologies can improve the quality of information about grid software, hardware, and data, enabling:
- the sharing of common understanding about domain concepts;
- reuse of domain knowledge;
- explicit definition of domain assumptions;
- interoperability enhancements.

We use the InteGrade opportunistic grid middleware as the base for a prototype implementation of our approach.

The Classical Paradigm

The so called Classical Paradigm is a modelling paradigm based on a set of characteristics appropriate to the open environment of the Semantic Web and Semantic Grid, e.g.:
- open world assumption;
- no adoption of the unique name assumption;
- inference support.

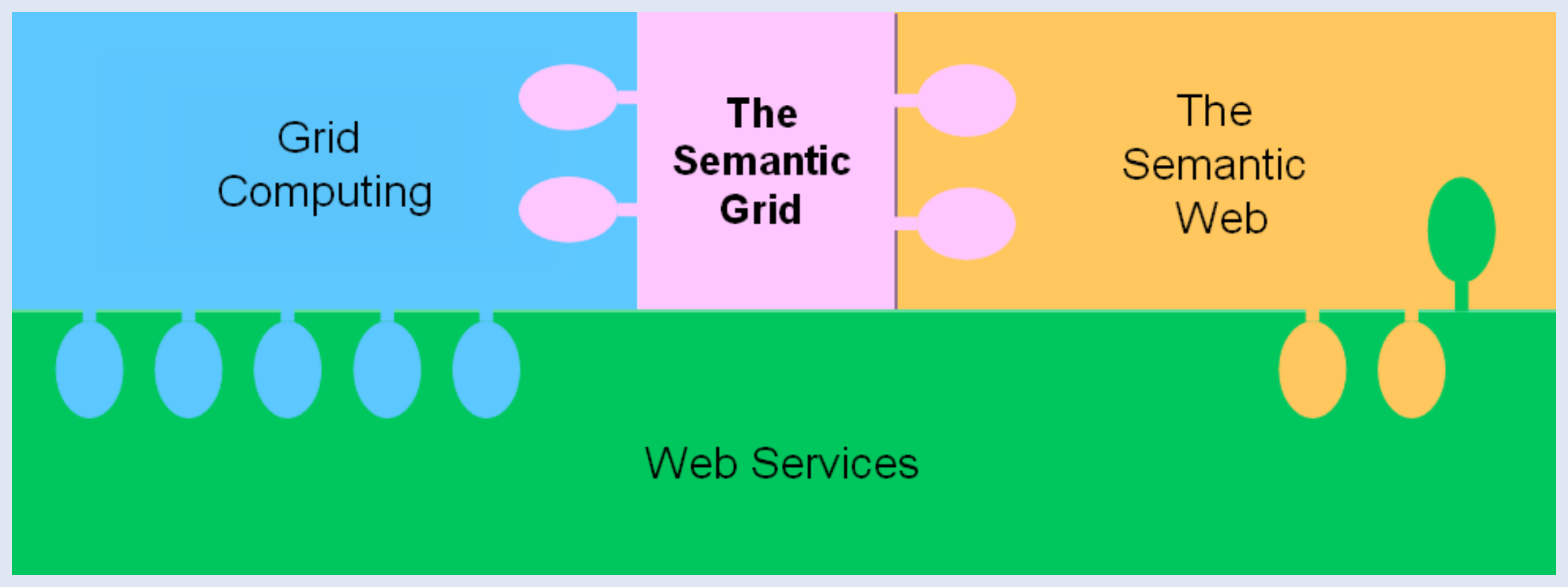


Figure 1: Grid Ontology Protégé-OWL

W3C-recommended ontology languages, such as OWL and RDF, are based on the classical Paradigm.

Defining a Grid Ontology

We started our work on defining the ontology by identifying the most basic elements present in the grid domain.

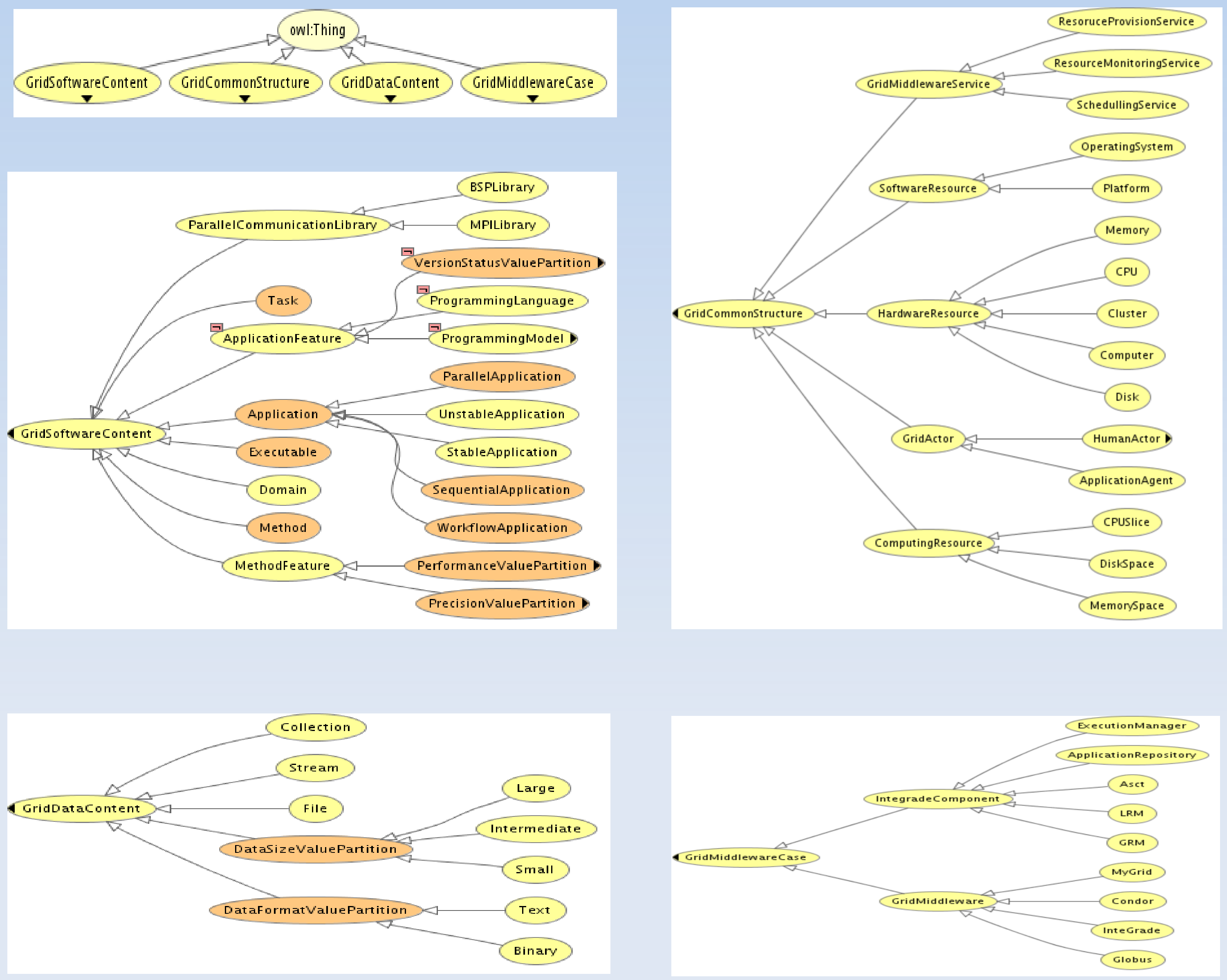
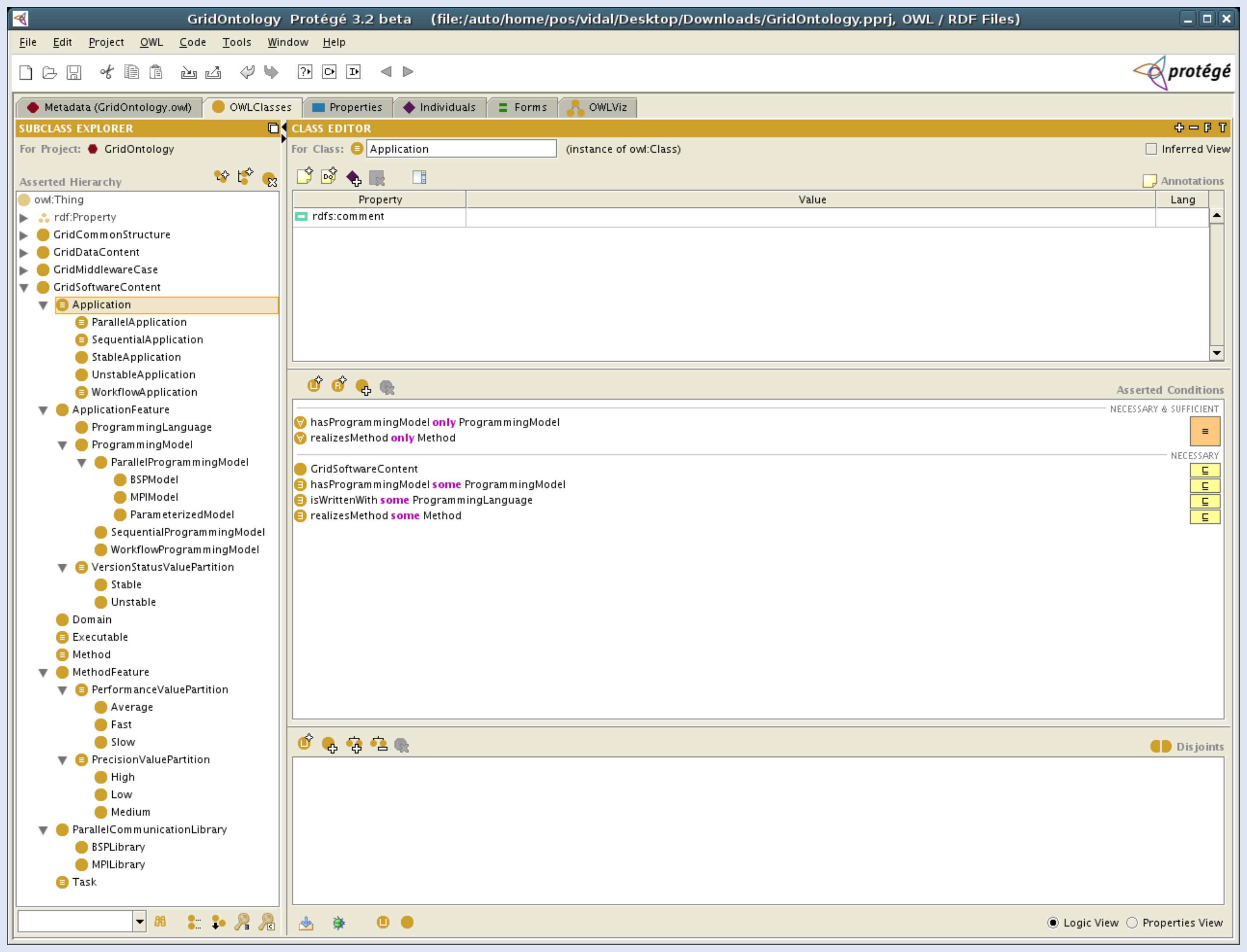


Figure 2: Grid Ontology Hierarchies

The Protégé-OWL tool enables the creation of an extensible OWL ontology. Reasoning to check consistency and to infer class hierarchy may be performed through reasoning tools such as Pellet.



For Project: GridOntology

For Class: Application (instance of owl:Class)

Property: rdfs:comment

```
owl:Thing
  rdf:Property
  GridCommonStructure
  GridDataContent
  GridMiddlewareCase
  GridSoftwareContent
    Application
      ParallelApplication
      SequentialApplication
      StableApplication
      UnstableApplication
      WorkflowApplication
    ApplicationFeature
      ProgrammingLanguage
      ProgrammingModel
        ParallelProgrammingModel
          BSPModel
          MPIModel
          ParameterizedModel
        SequentialProgrammingModel
        WorkflowProgrammingModel
      VersionStatusValuePartition
        Stable
        Unstable
    Domain
    Executable
    Method
    MethodFeature
      PerformanceValuePartition
        Average
        Fast
        Slow
      PrecisionValuePartition
        High
        Low
        Medium
    ParallelCommunicationLibrary
      BSPLibrary
      MPILibrary
    Task
```

```
hasProgrammingModel only ProgrammingModel
realizesMethod only Method

GridSoftwareContent
hasProgrammingModel some ProgrammingModel
isWrittenWith some ProgrammingLanguage
realizesMethod some Method
```

Figure 3: Grid Ontology Protégé-OWL

Exploring the Ontology

Initially, the ontology can act as a fundamental taxonomy encompassing the main concepts related to grid systems.

Grid users or the middleware itself can pose questions about the grid content by using this ontology, the knowledge base, query languages, and inference engines embedded in the middleware,

Conclusions not explicitly present in a knowledge base can be inferred from rules and axioms in this base.

Implementation Aspects

The key component in the architecture is the Grid Ontology Framework, which uses the Protégé-OWL API , an open-source Java library for OWL and RDF(S).

Grid middleware developers create and maintain the grid ontology.

Grid application developers define problem domains and their associated tasks, methods, and applications.

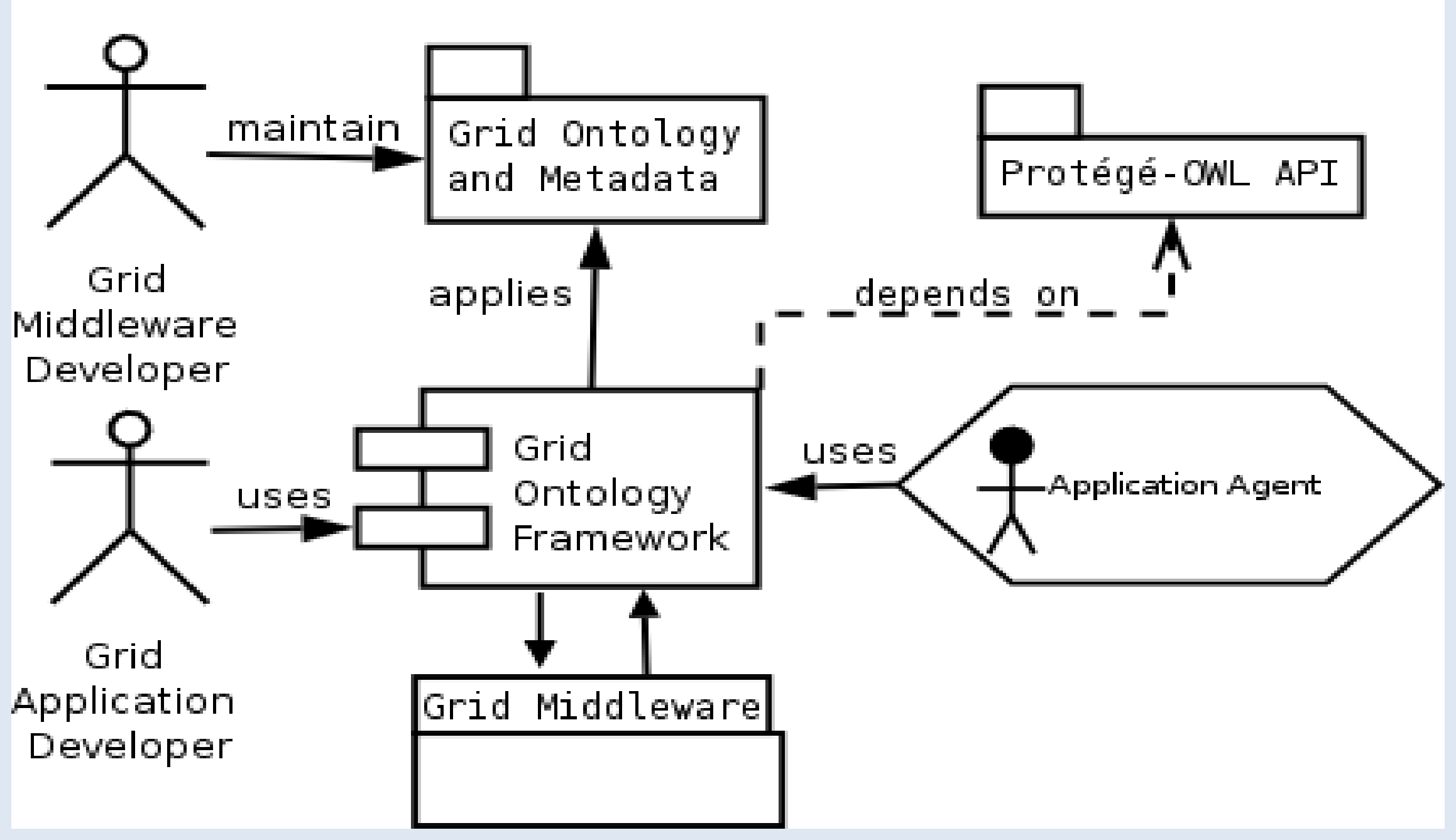


Figure 4: Grid System Architecture

Conclusion and Future Work

Using a framework connected to a grid ontology allowed us to explore the ontology, including support for ontology-based grid applications and improvements to grid resource usage.

We are investigating the use of the OWL query language (OWL-QL) features, such as:
- query-answering dialogues automated through reasoning methods;
- multiple distributed knowledge bases on the Semantic Web.