The Semantic Grid

Wei Xing[1] , Marios Dikaiakos[2]

[1]School of Computer Science
University of Manchester

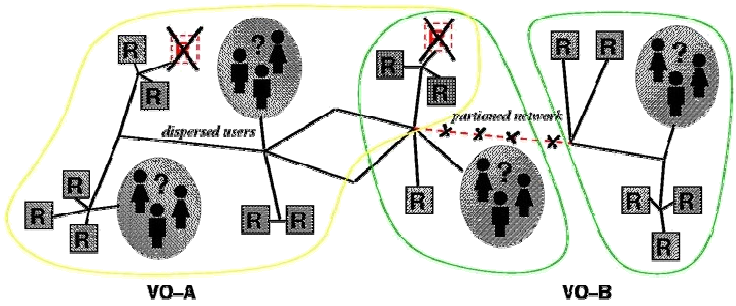
[2]Department of Computer Science
University of Cyprus

Outline

- Introduction
- Semantic Web 101
- A brief history of the Semantic Grid
- Semantics in the Grid
- A reference architecture for the Semantic Grid (S-OGSA)
- Next generation Semantic Grid (SOKU)
- Conclusions

What is Grid?

- The "Grid"
  - flexible, secure, coordinated resource sharing among dynamic collections of individuals, institutions, and resources - virtual organizations.



What is the Semantic Grid?

- An extension of the current Grid in which information and services are given well-defined and explicitly represented meaning, so that it can be shared and used by humans and machines, better enabling them to work in cooperation

Why we need the Semantic Grid?

Grid → “It is a truth universally acknowledged, that an application in possession of good middleware, must be in want of meaningful metadata.”

↑ Semantic

-- prof. C. Goble

Managing Metadata in Middleware

- Embedding and implicit metadata is the enemy of shareability and reuse in an open and decoupled and collaborative environment.
  - Expose it.
- Machine processable metadata is machine actionable metadata Enrich it.
  - With meaning (Semantics).

Don't we have Semantics in the Grid already?

- Its called metadata.
- Or vocabularies.
- Or glossaries.
- It's the state properties of a resource.
- Its in information services.
- And registries and catalogues.
- And configuration files.
- And policy definitions.
- And service level agreements.
- And file names.
- And file headers.
- And directory naming conventions
- And code libraries.
- And type systems.
- And schemas.
- And applications.
- And data formats.
- And best practice.
- And documentation.
- And workflows.
- And notification events
- And monitoring logs
- And embedded in XML tags …
- And even ontologies!
- And protocols.
- And decision procedures.

Outline

- Introduction
- Semantic Web 101
- A brief history of the Semantic Grid
- Semantics in the Grid
- A reference architecture for the Semantic Grid (S-OGSA)
- Next generation Semantic Grid (SOKU)
- Conclusions

What is the Semantic Web?

- "It's the Web of Data. Data is what's in databases. Imagine it's linked up like documents are linked up on the Web"
- "Imagine a spreadsheet where you can import data about anything from anywhere"
- "RDF is to data what HTML is to documents"

Need to Add "Semantics"

External agreement on meaning of annotations

- E.g., *Dublin Core* for annotation of library/bibliographic information
  - Agree on the meaning of a set of annotation tags
- Problems
  - Inflexib
  - Limited

Machine **Processable**
not
Machine **Understandable**

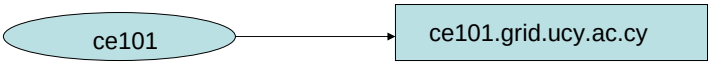
- Use Onto annotations
  - Ontologies provide a vocabulary of terms
  - New terms can be formed by combining existing ones
    - "Conceptual Lego"
  - Meaning (semantics) of such terms is formally specified
  - Can also specify relationships between terms in multiple ontologies

RDF

- RDF stands for Resource Description Framework
- It is a W3C Recommendation
  - http://www.w3.org/RDF
- RDF is a graphical formalism (+ XML syntax + semantics)
  - for representing metadata
  - for describing the semantics of information in a machine- accessible way
- Provides a simple data model based on triples.

The RDF Data Model

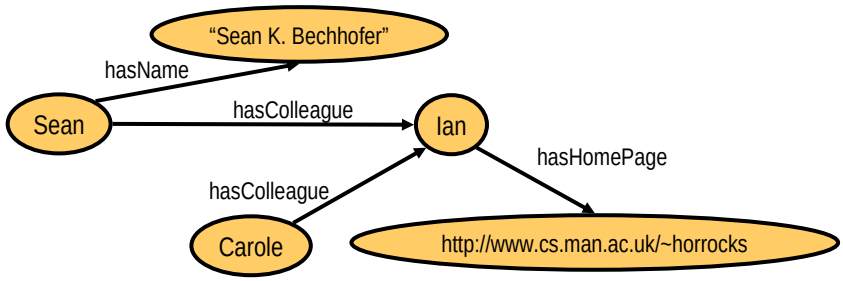
- Statements are <subject, predicate, object> triples:
  - <ce101,hasName,ce101.grid.ucy.ac.cy>
- Can be represented as a graph:



- Statements describe properties of resources
- A resource is any object that can be pointed to by a URI:
  - The generic set of all names/addresses that are short strings that refer to resources
  - a document, a picture, a paragraph on the Web, http://www.cs.man.ac.uk/index.html, a book in the library, a real person (?), isbn://0141184280
- Properties themselves are also resources (URIs)

Linking Statements

- The subject of one statement can be the object of another
- Such collections of statements form a directed, labeled graph



- The object of a triple can also be a "literal" (a string)

RDF Syntax

- RDF has an XML syntax that has a specific meaning:
- Every Description element describes a resource
- Every attribute or nested element inside a Description is a property of that Resource
- We can refer to resources by URIs

```
<rdf:Description rdf:about="some.uri/person/sean_bechhofer">
  <o:hasColleague resource="some.uri/person/ian_horrocks"/>
  <o:hasName rdf:datatype="&xsd;string">Sean K. Bechhofer</o:hasName>
</rdf:Description>
<rdf:Description rdf:about="some.uri/person/ian_horrocks">
  <o:hasHomePage>http://www.cs.mam.ac.uk/~horrocks</o:hasHomePage>
</rdf:Description>
<rdf:Description rdf:about="some.uri/person/carole_goble">
  <o:hasColleague resource="some.uri/person/ian_horrocks"/>
</rdf:Description>
```

What does RDF give us?

- A mechanism for **annotating** data and resources.
- Single (simple) data model.
- Syntactic consistency between names (URIs).
- Low level **integration** of data.

RDF(S): RDF Schema

- RDF gives a formalism for meta data annotation, and a way to write it down in XML, but it does not give any special meaning to vocabulary such as subClassOf or type (supporting OO-style modelling)
- RDF Schema extends RDF with a schema vocabulary that allows you to define basic vocabulary terms and the relations between those terms
  - Class, type, subClassOf,
  - Property, subPropertyOf, range, domain
  - it gives “extra meaning” to particular RDF predicates and resources
  - this “extra meaning”, or semantics, specifies how a term should be interpreted

Problems with RDFS

- RDFS is too weak to describe resources in sufficient detail
  - No localised range and domain constraints
    - Can't say that the range of hasChild is person when applied to persons and elephant when applied to elephants
  - No existence/cardinality constraints
    - Can't say that all *instances* of person have a mother that is also a person, or that persons have exactly 2 parents
  - No transitive, inverse or symmetrical properties
    - Can't say that isPartOf is a transitive property, that hasPart is the inverse of isPartOf or that touches is symmetrical
- It can be difficult to provide reasoning support
  - No "native" reasoners for non-standard semantics

Ontology in Computer Science

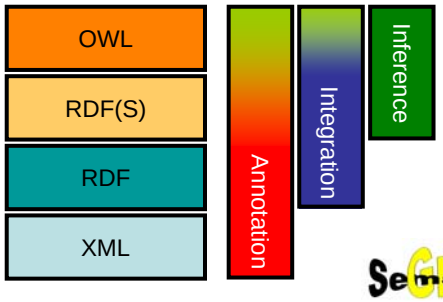
- An ontology is an **engineering artifact**:
  - It is constituted by a specific **vocabulary** used to describe a certain reality, plus
  - a set of explicit **assumptions** regarding the intended meaning of the vocabulary.
    - Almost always including how concepts should be **classified**
- Thus, an ontology describes a formal specification of a certain domain:
  - Shared understanding of a domain of interest
  - Formal and **machine manipulable** model of a domain of interest

Building a Semantic Web

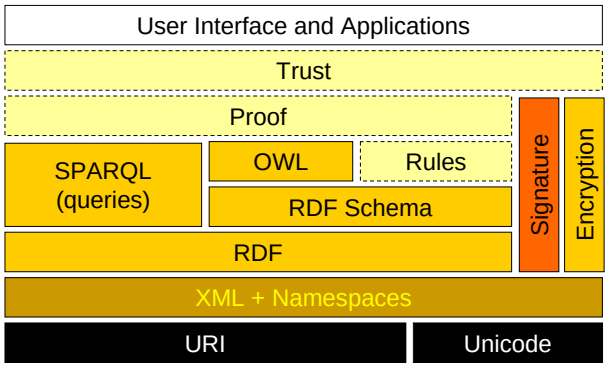
- Annotation
  - Associating metadata with resources
- Integration
  - Integrating information sources
- Inference
  - Reasoning over the information we have.
  - Could be light-weight (taxonomy)
  - Could be heavy-weight (logic-style)
- Interoperation and Sharing are key goals

Languages

- Work on Semantic Web has concentrated on the definition of a collection or “stack” of languages.
  - These languages are then used to support the representation and use of metadata.
- The languages provide basic machinery that we can use to represent the extra semantic information needed for the Semantic Web
  - XML
  - RDF
  - RDF(S)
  - OWL
  - …



The Semantic Web layer cake



Attribution

Explanation

Ontologies + Inference

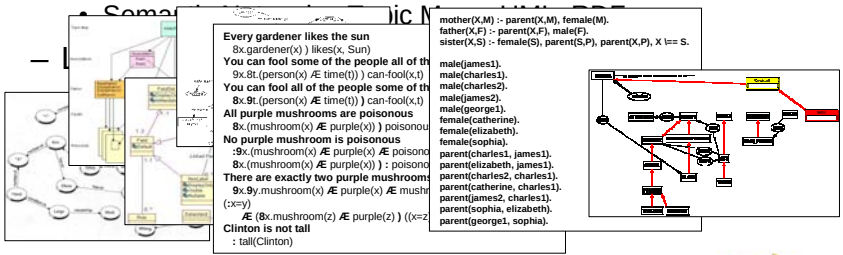
Metadata

Standard syntax

Identity

Ontology Languages

- We need languages that allow us to represent this information
  - Ontology Languages!
- There are a wide variety of languages for this "Explicit Specification"
  - Graphical



OWL

- W3C Recommendation (February 2004)
- Well defined RDF/XML serializations
- A family of Languages
  - OWL Full
  - OWL DL
  - OWL Lite
- Formal semantics
  - First Order (DL/Lite)
  - Relationship with RDF
- Comprehensive test cases for tools/implementations
- Growing industrial takeup.

OWL Basics

- Set of constructors for concept expressions
  - Booleans: and/or/not
  - Quantification: some/all
- Axioms for expressing constraints
  - Necessary and Sufficient conditions on classes
  - Disjointness
  - Property characteristics: transitivity, inverse
- Facts
  - Assertions about individuals

Reasoning with OWL

- OWL (DL) has a well defined semantics that tells us how to interpret expressions in the language.
- This semantics corresponds to “traditional” interpretations given to first order logic or subsets of FOL like Description Logics.
- OWL DL based on a well understood Description Logic Formal properties well understood (complexity, decidability)
  - Known reasoning algorithms
  - Implemented systems (highly optimised)
- Because of this, we can reason about OWL ontologies, allowing us to draw inferences from the basic facts that we provide.

Why Reasoning?

- Reasoning can be used as a design support tool
  - Check logical consistency of classes
  - Compute implicit class hierarchy
- May be less important in small local ontologies
  - Can still be useful tool for design and maintenance
  - Much more important with larger ontologies/multiple authors
- Valuable tool for integrating and sharing ontologies
  - Use definitions/axioms to establish inter-ontology relationships
  - Check for consistency and (unexpected) implied relationships
- Basis for answering queries.
- Reasoning can help underpin the provision of the machine processing required of the Semantic Web.

What does OWL give us?

- Rich language for describing domain models.
- **Unambiguous** interpretations of complex descriptions.
- The ability to use **inference** to manage our vocabularies.

Java tools for Semantic Web Technology (1)

- RDF tools
  - Jena
    - a Java framework for building Semantic Web applications.
    - It provides a programmatic environment for RDF, RDFS and OWL, SPARQL and
    - A rule-based inference engine.
  - OpenRDF (aka. Sesame)
    - Sesame is an open source framework for storage, inferencing and querying of RDF data.
    - Sesame RQL, and SPARQL
  - SPARQL
    - Query Language for RDF
    - By RDF Data Access Working Group
    - A W3C Candidate Recommendation

Java tools for Semantic Web Technology (2)

- OWL
  - Protégé
    - An ontology editor and knowledge-base framework
  - OWL API
    - Java OWL API (OWL1.0 and OWL 1.1)
  - Pellet
    - OWL DL reasoner in Java
  - FaCT++
    - OWL DL reasoner in C++

Outline

- Introduction
- Semantic Web 101
- A brief history of the Semantic Grid
- Semantics in the Grid
- A reference architecture for the Semantic Grid (S-OGSA)
- Next generation Semantic Grid (SOKU)
- Conclusions

The Semantic Grid Report 2001

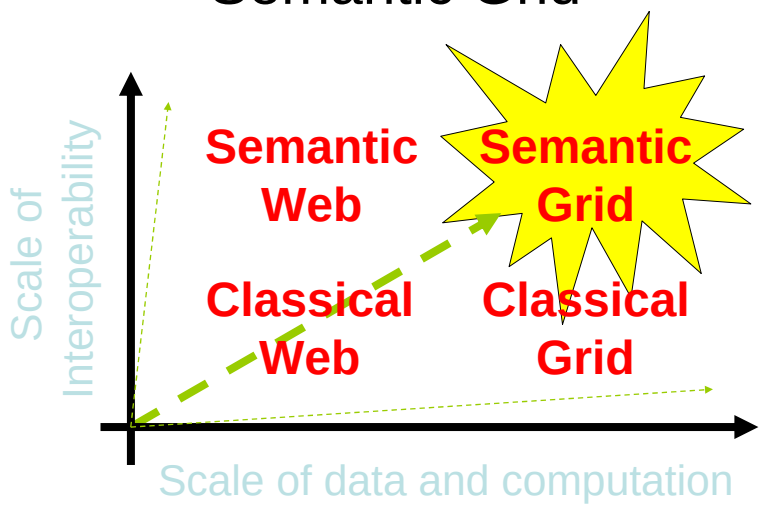
- At this time, there are a number of grid applications being developed and there is a whole raft of computer technologies that provide fragments of the necessary functionality.
- However there is currently a major gap between these endeavours and the vision of e-Science in which there is a high degree of easy-to-use and seamless automation and in which there are flexible collaborations and computations on a global scale.

www.semanticgrid.org

Report updated – March 2005 issue of Proceedings of the IEEE

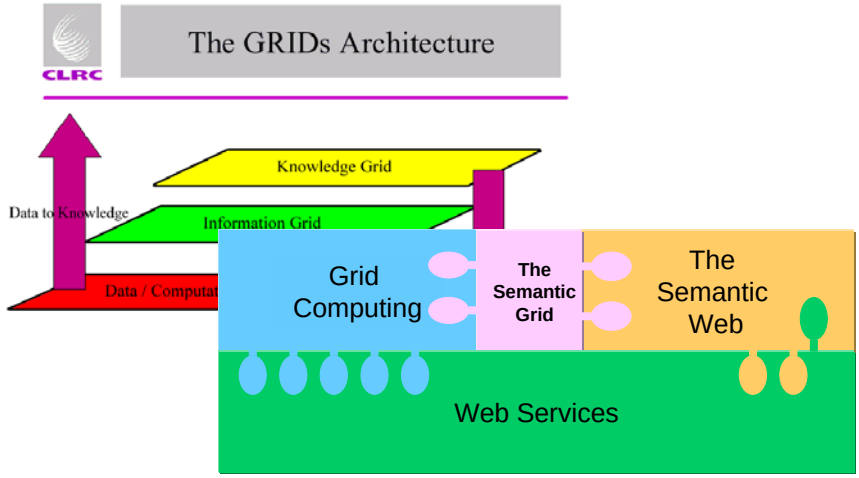
Building bridges

Semantic Grid



Based on an idea by Norman Paton

Semantics in and on the Grid



Outline

- Introduction
- Semantic Web 101
- A brief history of the Semantic Grid
- Semantics in the Grid
- A reference architecture for the Semantic Grid (S-OGSA)
- Next generation Semantic Grid (SOKU)
- Conclusions

What?

- An extension of the Grid
- Rich metadata is exposed and handled explicitly, shared, and managed via Grid protocols

How?

- The Semantic Grid uses metadata to describe information in the Grid.
- Turning information into something more than just a collection of data means understanding the context, format, and significance of the data.
- Therefore:
  - Understand information
  - Discovery and reuse

Semantic?

- Semantic = metadata + meaning
- Metadata explicitly exposed as a first class object in a machine processable form.
- Controlled vocabularies or knowledge models (aka Ontologies) for describing metadata in a machine processable form.
- Schemas for structuring metadata in a machine processable form.
- Rules over metadata.

Possibly using Semantic Web technologies
For people and machines

Metadata Sharing and Reusing

- If semantics is embedded or closely coupled
  - Its hard to adapt
- If its represented in different formats
- If its created and used and destroyed using different protocols and mechanisms
  - Its hard to share
  - Its hard to reuse
  - Its hard to reinterpret

Requirements of the Semantic Grid

- Systematic management of metadata in middleware
  - the creation, update, query metadata
- Semantic enrichment of metadata in middleware
  - Machine processable metadata is machine actionable metadata

Building a Semantic Grid

- how a Grid might be developed or adapted in a way that would allow other people to make use of the resources you are looking to provide.
  - Common standard information model
  - Semantic-able
- how to describe and define the data or resources that will be stored and used by a Grid.
  - Domain knowledge
  - Description Logic

Use Cases

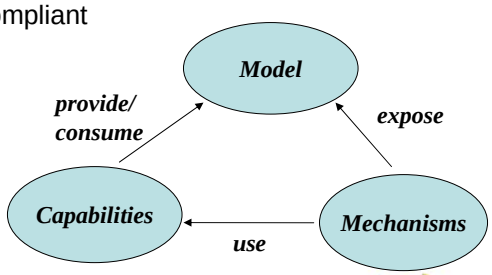
- Semantic Grid for Annotation of Data
  - Already seen before in the cases of BioPAX and Gene Ontology
- Semantic Grid in Workflows
  - Service description and discovery (myGrid)
- Semantic Grid in Data Integration
  - Data Integration (www.godatabase.org)
  - Data Integration (GEON)
- Semantic Grid in Authorisation
  - We will see an example later

Outline

- Introduction
- Semantic Web 101
- A brief history of the Semantic Grid
- Semantics in the Grid
- A reference architecture for the Semantic Grid (S-OGSA)
- Next generation Semantic Grid (SOKU)
- Conclusions

S-OGSA (1)

- Semantic-OGSA (S-OGSA) is...
  - Our proposed Semantic Grid reference architecture
  - A low-impact extension of OGSA
    - Mixed ecosystem of Grid and Semantic Grid services
      - Services ignorant of semantics
      - Services aware of semantics but unable to process them
      - Services aware of semantics and able to process (part of) them
    - Everything is OGSA compliant
  - Defined by
    - Information model
      - New entities
    - Capabilites
      - New functionalities
    - Mechanisms
      - How it is delivered



S-OGSA (2)

- How to provide:
  - Just give the semantic metadata to those services
  - Or we can have the semantic services by SOGSA own.
- There are no big differences…
  - if the service can understand semantic (e.g., they support semantic API), then itself can be a S-OGSA service.

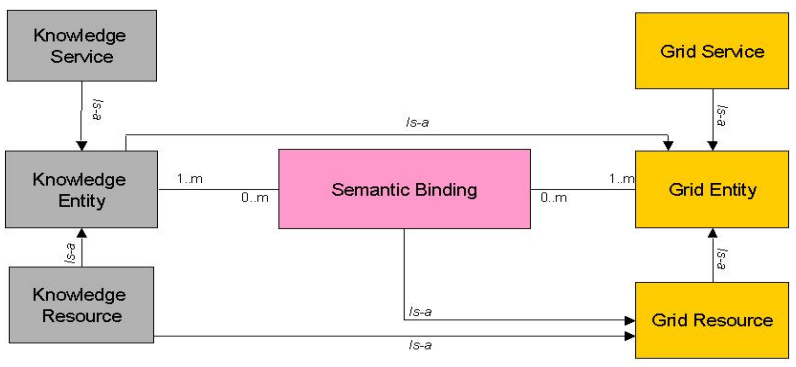
S-OGSA (3)

- A Grid usually consist of several different services by OGSA:
  - VO management service
  - Resource discovery and Management service
  - Job Management service
  - Security service
  - Data Management service
- The S-OGSA should (will) provide the metadata +semantic services to those services.

S-OGSA (4)

- The Solution:
  - Attached the semantic to Grid entities.
  - Binding them together by semantic binding service.
  - Normal grid services can be “semantic” by the semantic binding service.

S-OGSA Model. Semantic Bindings

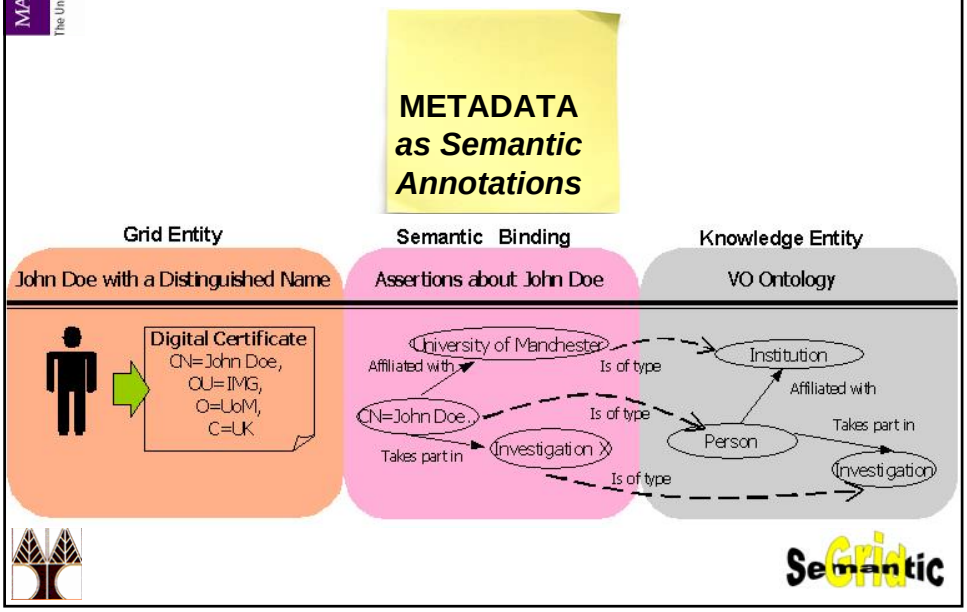


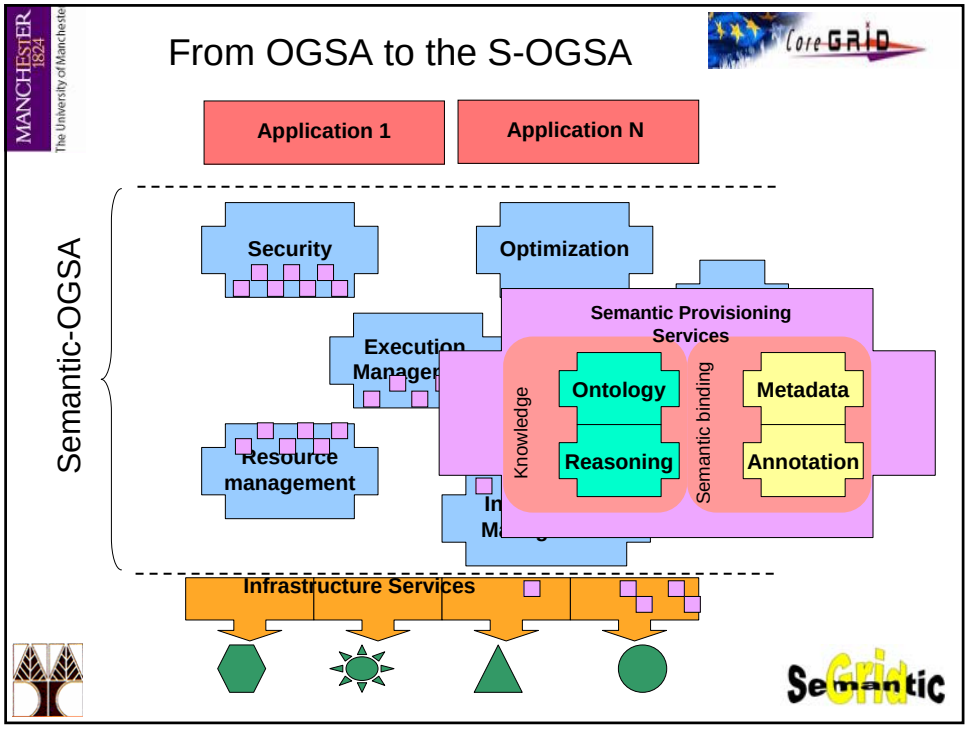
S-OGSA services

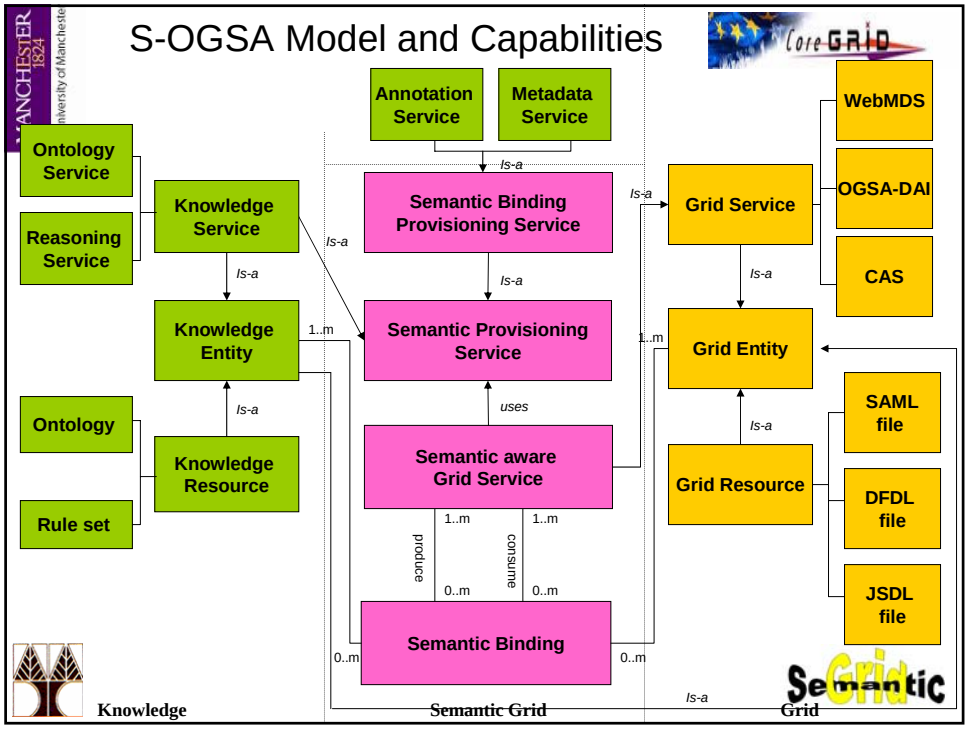
- Core OGSA-compliant Semantic Services
  - Semantic provisioning services like ontologies, semantic annotation services, semantic encoding services, metadata repositories, decision making services
  - Robust and scalable, capable of dealing with distribution
- Knowledge aware & enabled Grid Services
  - Re-factoring Grid services to be knowledge consumers and suppliers
  - Migration methodologies and mixed ecosystems
- Core knowledge content
  - Grid resource ontology, Application content

- OntoGrid http://www.ontogrid.net

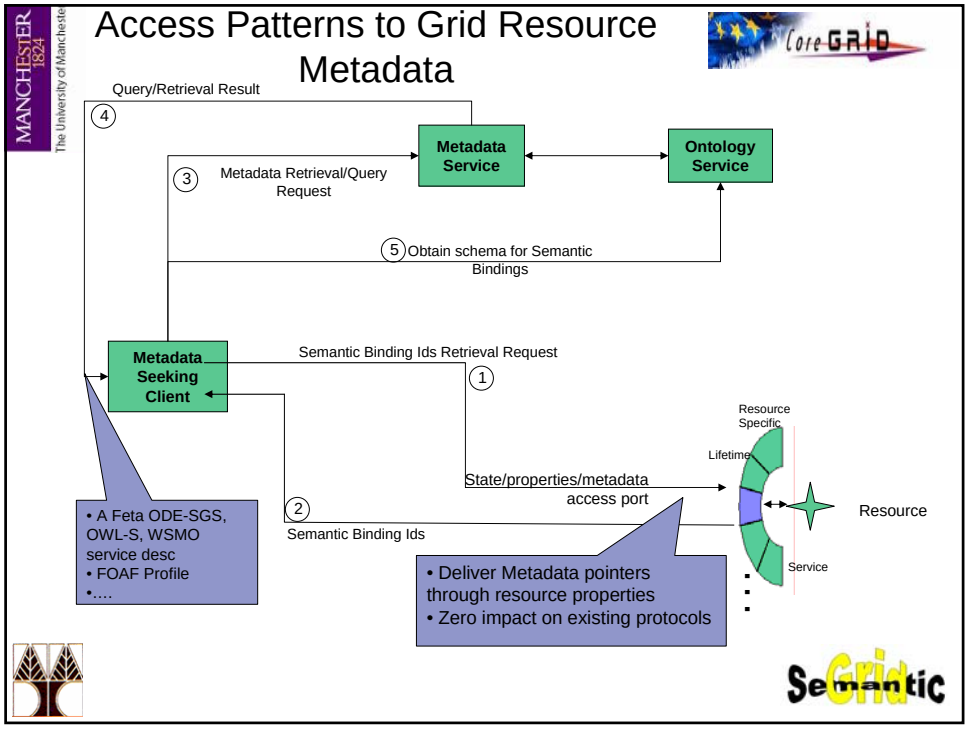
S-OGSA Model Example




From OGSA to the S-OGSA
Application 1
Application N
Semantic-OGSA
Security
Optimization
Execution Manager
Resource management
Semantic Provisioning Services
Knowledge
Ontology
Reasoning
Semantic binding
Metadata
Annotation
Infrastructure Services


S-OGSA Model and Capabilities
Annotation Service
Metadata Service
Ontology Service
Reasoning Service
Knowledge Service
Semantic Binding Provisioning Service
Grid Service
WebMDS
OGSA-DAI
CAS
Knowledge Entity
Semantic Provisioning Service
Grid Entity
Ontology
Knowledge Resource
Semantic aware Grid Service
Grid Resource
SAML file
DFDL file
JSDL file
Rule set
produce
consume
Semantic Binding
Knowledge
Semantic Grid
Grid
is-a
uses
1..m
0..m

Access Patterns to Grid Resource Metadata



The Semantic Grid Middleware

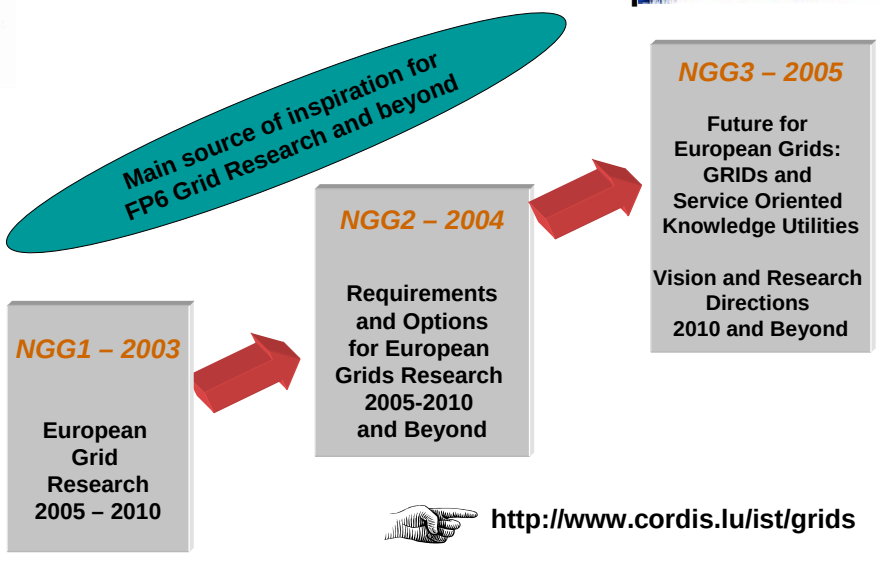
- The Knowledge entity
  - Core Grid Ontology (CGO)
    - Grid concepts, and their relationships defined in the ontology
- S-OGSA middleware services
  - Semantic Binding service
    - Semantic metadata management
  - S-OGSA-DAI: Semantically Aware Data Service
    - expose Semantic Bindings (SB) and accept RDF queries
    - improved information discovery and semantic Integration of heterogeneous data sources
  - Active Ontology
    - Semantic information integration service
    - dynamic, distributed information sources.

- **Ontogrid CVS**: http://www.ontogrid.net/ontogrid/downloads.jsp

Outline

- Introduction
- Semantic Web 101
- A brief history of the Semantic Grid
- Semantics in the Grid
- A reference architecture for the Semantic Grid (S-OGSA)
- Next generation Semantic Grid (SOKU)
- Conclusions

Next Generation Grids Reports



http://www.cordis.lu/ist/grids

NGG3

Next Generation Grids Report 2005

Future for European Grids: GRIDs and Service Oriented Knowledge Utilities – Vision and Research Directions 2010 and Beyond, December 2006

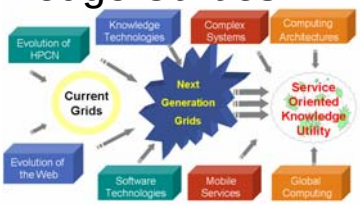
Service-Oriented Knowledge Utility (SOKU)

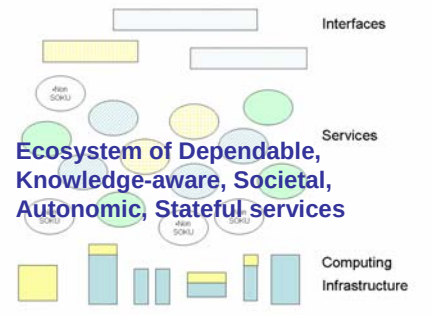
A flexible, powerful and cost-efficient way of building, operating and evolving IT intensive solutions for business, science and society.

- **Building on existing industry practices and emerging technologies**
- **Support ecosystems that promote collaboration and self-organisation**
- **Towards increased agility, lower cost, broader availability of services**
- **Empowering service providers, integrators and consumers of ICT**
- **(R)evolution of concepts from Web, Grid & Knowledge technologies**
- **Safe, ease and ubiquitous as existing utilities like electricity or water**

Service Oriented Knowledge Utilities

- Next Generation Grids Expert Group Report 3 (NGG3) published January 2006
- Converged vision of Next Generation Grids and Service Oriented Knowledge Utilities



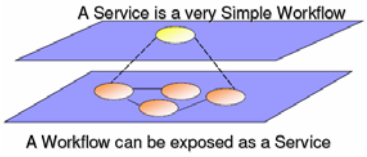


Ecosystem of Dependable, Knowledge-aware, Societal, Autonomic, Stateful services

- ***Service Oriented*** –services may be instantiated and assembled dynamically
- ***Knowledge*** –knowledge-assisted to facilitate automation, and processing and delivering knowledge
- ***Utility*** –directly and immediately useable service with established functionality, performance and dependability

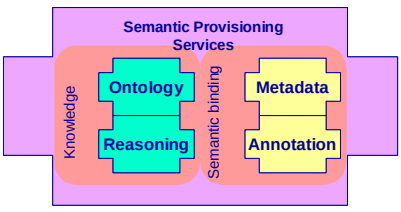
What is a SOKU service?

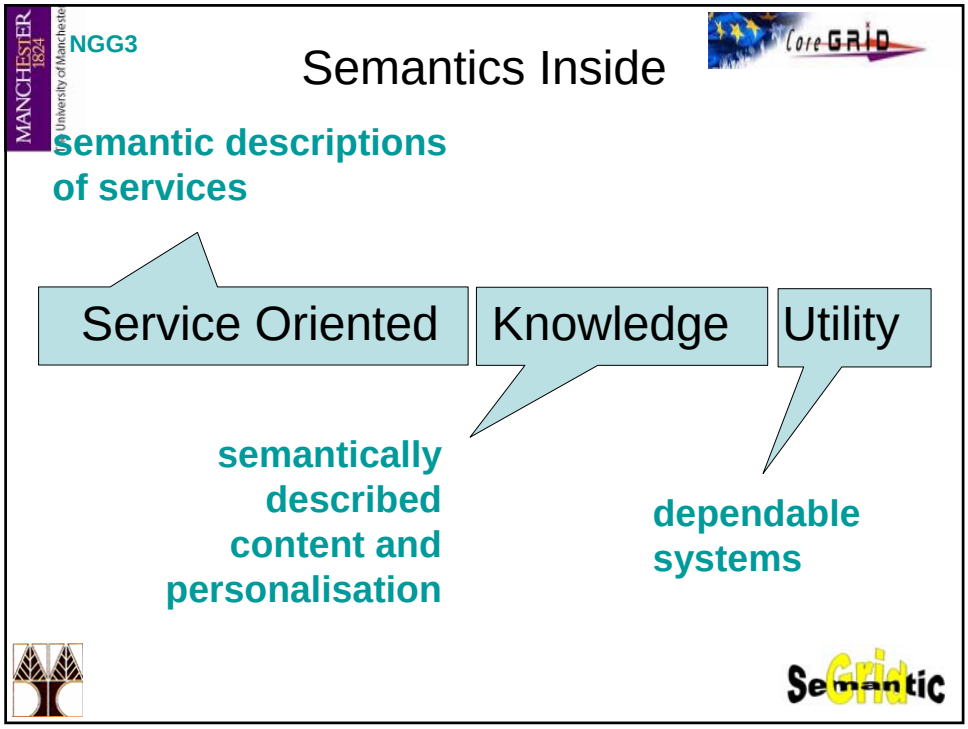
- SOKU services are *semantically described*, i.e. annotated with machine-processable metadata which facilitates their automated use.
  - Can be dynamically composed and configured
  - Adapt automatically, providing self-management and autonomic behaviour

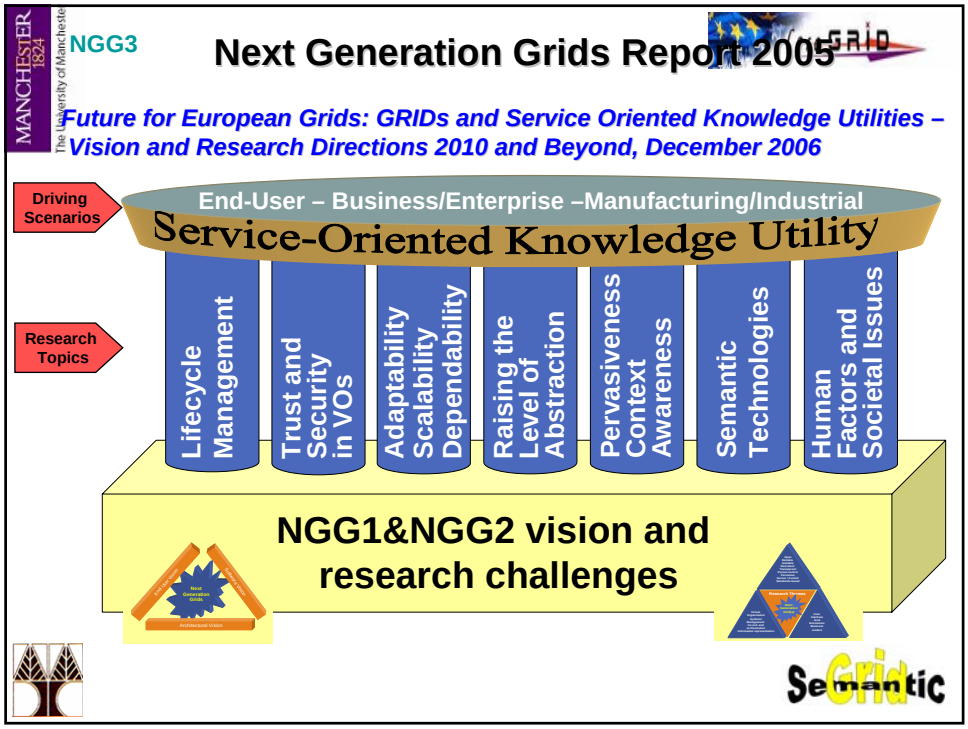


What is a SOKU service?

- SOKU services also work with semantically described content and semantic descriptions, i.e. they *process knowledge*
  - may contain and use it, consume it, or produce it




NGG3
Semantics Inside
semantic descriptions of services
Service Oriented
Knowledge
Utility
semantically described content and personalisation
dependable systems


NGG3
Next Generation Grids Report 2005
Future for European Grids: GRIDs and Service Oriented Knowledge Utilities – Vision and Research Directions 2010 and Beyond, December 2006
Driving Scenarios
End-User – Business/Enterprise –Manufacturing/Industrial
Service-Oriented Knowledge Utility
Research Topics
Lifecycle Management
Trust and Security in VOs
Adaptability Scalability Dependability
Raising the Level of Abstraction
Pervasiveness Context Awareness
Semantic Technologies
Human Factors and Societal Issues
NGG1&NGG2 vision and research challenges

Challenges

- Making it easier not harder
  - Avoid baroque architectures
  - A little bit of semantics goes a long way
- Acquiring knowledge.
  - Fostering network effects
  - Data generated in Semantic Data Web ready format from legacy resources.
  - Leveraging social tagging and automated tagging.
  - Simple content and plenty of it is better than clever content but poor coverage.
- Knowledge technologies that are scalable and robust.
- Semantic mechanisms invisible to people
- Semantic infrastructure visible to middleware
- Return on investment

More Information

- http://www.semanticgrid.org
- http://www.ontogrid.net

Acknowledgements

- Carole Goble
- David De Roure
- Sean Bechhofer
- Oscar Corcho

Ontogrid Team