

# S-MDS: A Semantic Monitoring and Discovery System for the Grid

Yusuke Tanimura  
National Institute of AIST

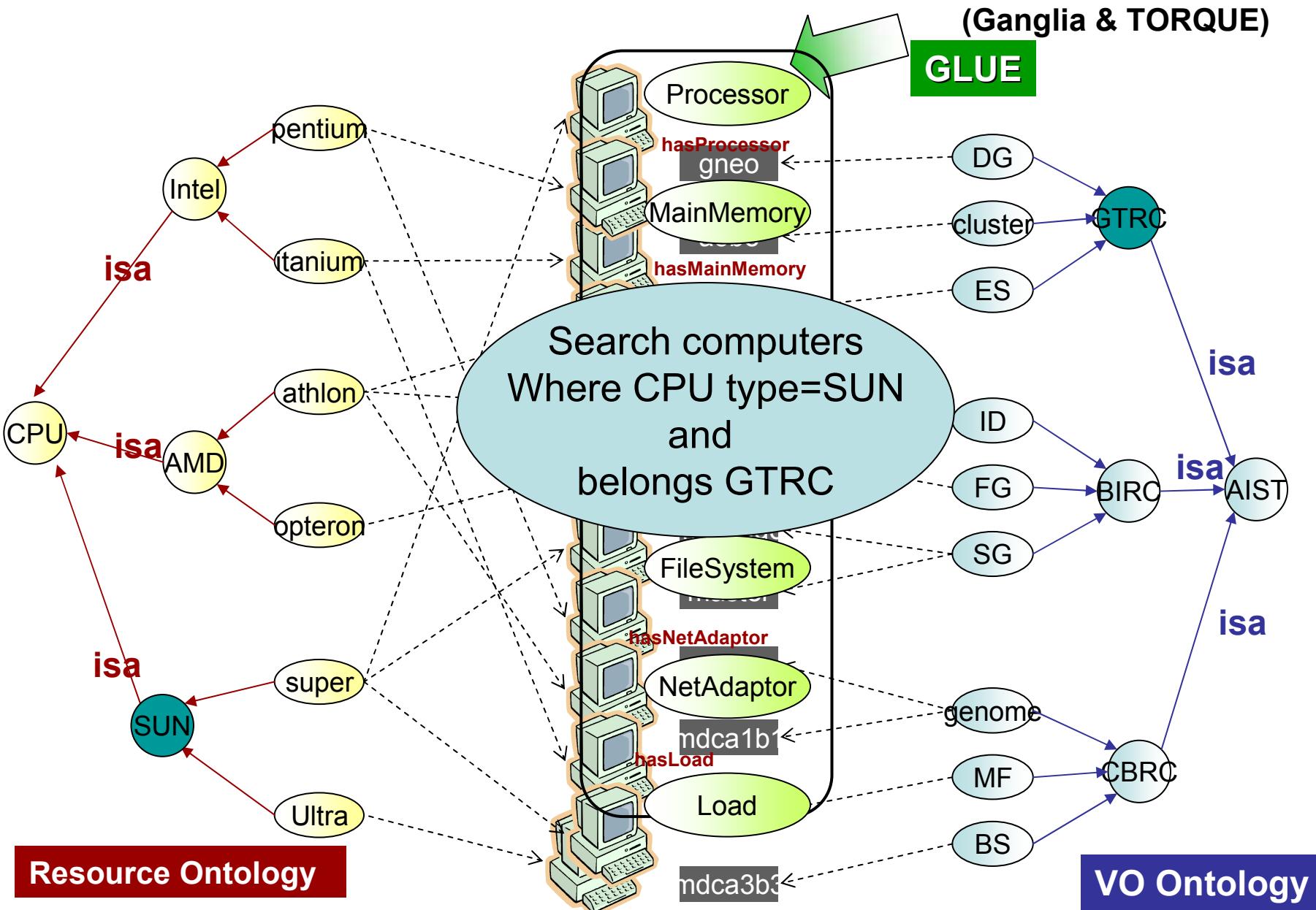
# Outline

- S-MDS Introduction
- Setting up/Using S-MDS
  - Demos
- Recent Functional Enhancements

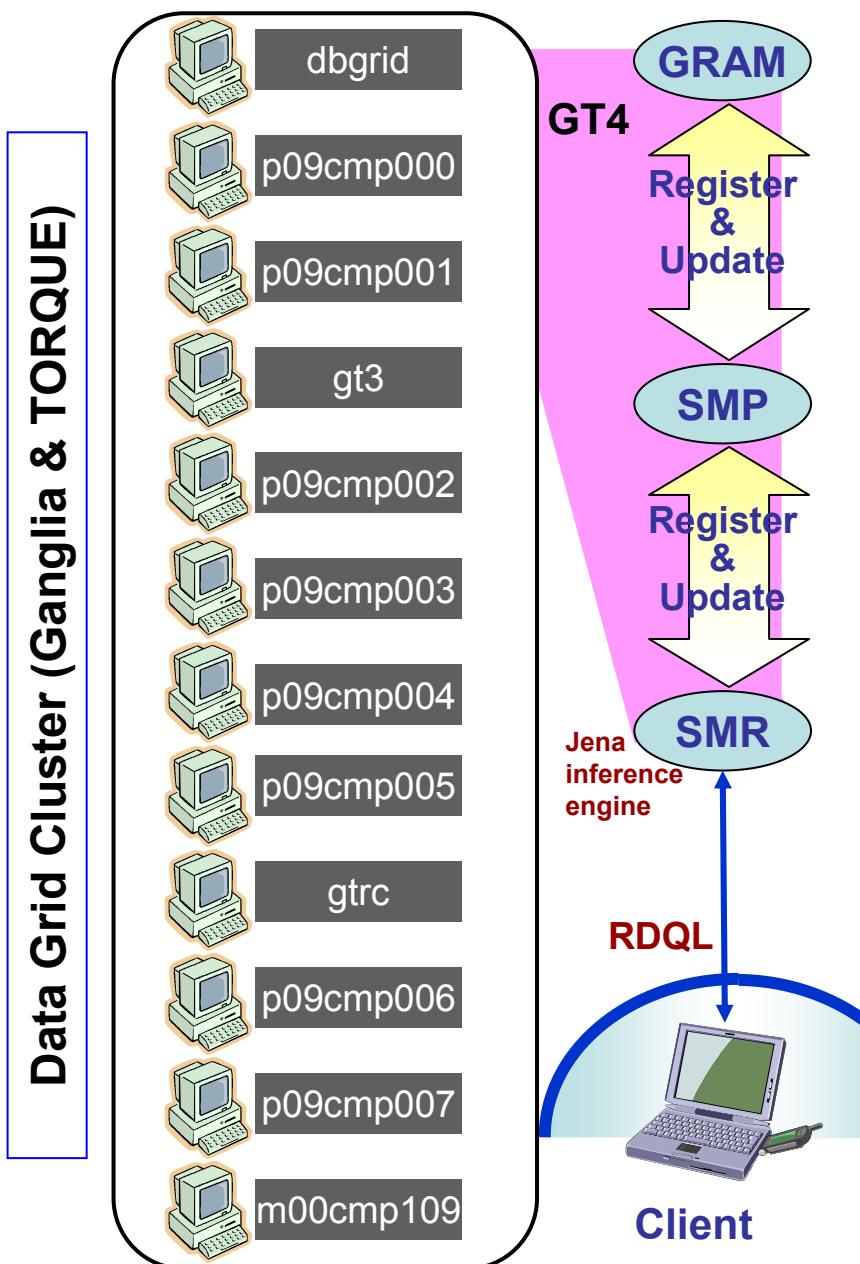
# What does S-MDS do? (1)

- You can define **Ontologies** over current monitoring system (Ganglia, TORQUE, etc.).
  - Ontology: a set of concepts within a domain and the relationships between those concepts
    - Ex. Athron and Pentium is both Intel-architecture.
    - Ex. GTRC and ITRI both belongs AIST organization.
- Discover the resource using a **Query Language SPARQL**.
  - Ex. Search intel-architecture computers where current load average is under 1 and belong AIST.

# Ex. Grid Resource Discovery Using Ontology



# What does S-MDS do (2) ?



- Monitor the resource **continually**
  - Ex. Search loadavg of intel-architecture (monitor mode)

The following graph shows history of the CPU load average and StdDev, which were retrieved by continuous query.



# Resource Properties

- Resource/service state is described by Resource Properties (RPs).
  - GetResourceProperty etc.
- **Problem:** deal only with the document structures but NOT with its contained semantics/concept!!

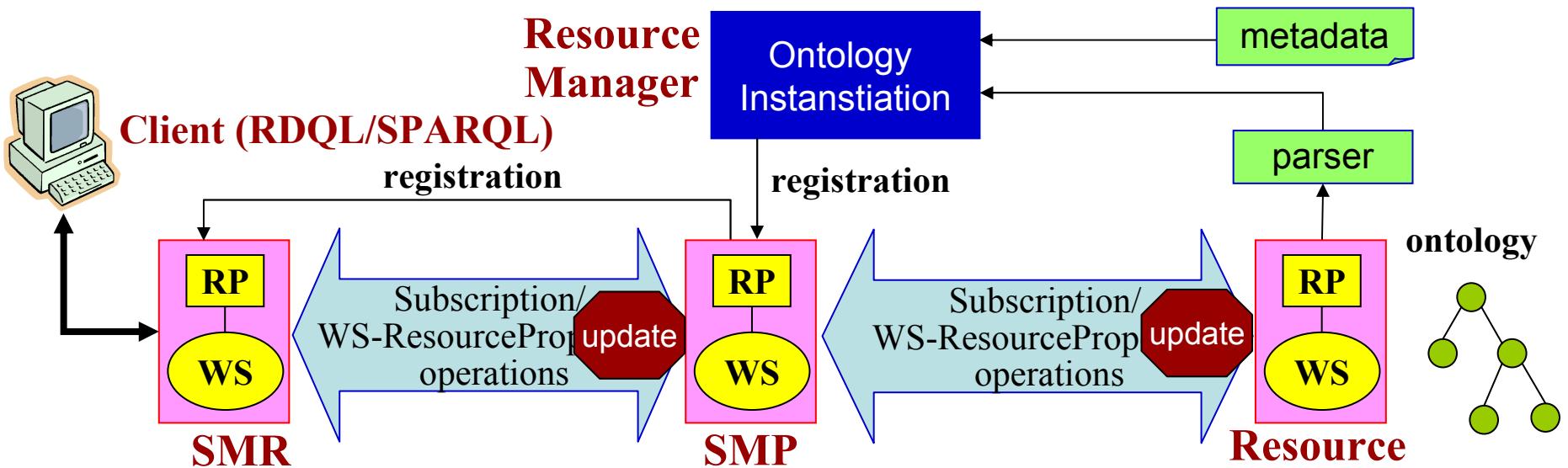
# Querying Problems

```
<Cluster Name="datagrid" UniqueID="datagrid">
  <SubCluster Name="datagrid" UniqueID="datagrid">
    <Host Name="dbgrid.x.x" UniqueID="dbgrid.x.x">
      <Processor CacheL1="0" CacheL1D="0" CacheL1I="0"
                 CacheL2="0" ClockSpeed="3001" InstructionSet="x86"/>
      <MainMemory RAMAvailable="19" RAMSize="1484"
                    VirtualAvailable="332" VirtualSize="2041"/>
      <OperatingSystem Name="Linux"
                        Release="2.6.0-test5_2smp"/>
      <Architecture SMPSize="2"/>
      <FileSystem AvailableSpace="65511" Name="entire"
                   ReadOnly="false" Root="/" Size="82459"/>
      <NetworkAdapter IPAddress="163.220.2.54"
                        InboundIP="true" MTU="0" Name="dbgrid.hpcc.jp"
                        OutboundIP="true"/>
      <ProcessorLoad Last15Min="10" Last1Min="6"
                     Last5Min="8"/>
    </Host>
    <Host/>
  ...
</SubCluster>
</Cluster>
```

- Processable queries
  - Hosts with x86 processor instruction set
    - //Host[Processor@InstructionSet='x86']
  - Host with Linux OS
    - //Host[OperatingSystem[@Name='Linux']]
- Non-processable queries
  - Hosts with intel compatible architecture
  - Hosts with Unix compatible OS

# S-MDS

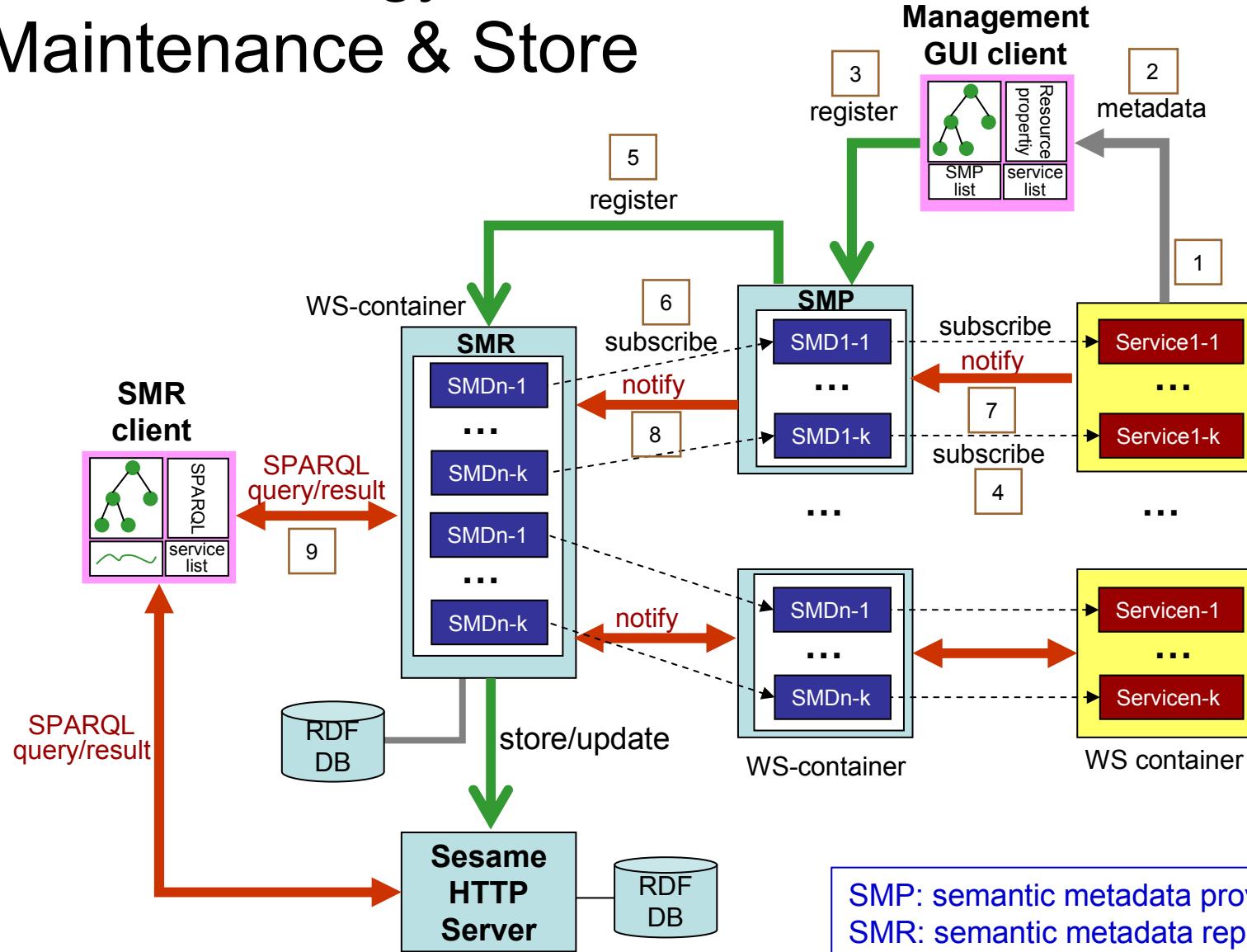
- Define **an ontology** for each service domain.
  - Resource ontology, VO ontology, GRAM ontology, DAI ontology, ...
- Map resource property values into the ontology classes.
- Semantic Metadata Provider (**SMP**): maintain ontology instance (semantic metadata)
- Semantic Metadata Repository (**SMR**): instance aggregation & query



# What does S-MDS do? (3)

- Describe the resource properties (RP) using ontology: RP  $\leftrightarrow$  Ontology mapping.
- Store and maintain the ontology (instances) in a WS or ontology (RDF) repository so that it can be semantically queried.
  - SPARQL or RDF query language (with inference)
- Provide a GUI for easy SPARQL creation.

# Ontology Maintenance & Store



# Setting Up/Using S-MDS

- Problem & Motivation

**Grid user is not an expert of the Semantic Web.**

-> Provide GUI-based Support Tools

1. I don't want to describe OWL (Web Ontology Language) for my cluster resource.

Provide Automatic Ontology Mapping Tool

- Convert your ***ResourceProperty*** to ***OWL Ontology***
- Easy to add your ***custom ontology***

2. I don't want to write an SPARQL query for monitoring.

Provide GUI-based Query & Monitoring Tool

- Just view the properties and give conditions

# S-MDS: GUI-based Resource Monitoring and Access

SPARQL Builder

SMR EPR(End Point Reference)

Container URL: <http://dbgrid.hpcc.jp:7373>

Address: <http://163.220.2.54:7373/wsrf/services/s-mds/SmrService>

TW List: 2. <http://163.220.2.54:8080/wsrf/services/ManagedJobFactoryService - ResourceID-PBS>

\* SPARQL Query Settings Saved File:

Domain Specific Ontology Schema

- ns24:NonRootElement
- ns24:Architecture
- ns24:Processor
  - ns24:hasSemantics ( CpuType )
    - hasType Itanium
  - ns24:hasProcessorCacheL1I
  - ns24:hasProcessorCacheL1
  - ns24:hasProcessorInstructionSet
  - ns24:hasProcessorCacheL2
  - ns24:hasProcessorCacheL1D
  - ns24:hasProcessorClockSpeed
- ns24:Policy
- ns24:ProcessorLoad
- ns24:NetworkAdapter
- ns24:OperatingSystem
  - ns24:hasOperatingSystemRelease
  - ns24:hasOperatingSystemName
    - = Linux
- ns24:Info
- ns24:MainMemory
- ns24:FileSystem
- ns24:Host
  - ns24:hasSemantics ( AIST )
  - hasType EScience

Set Type For \*ns24:hasSemantics ( CpuType )

hasType Itanium

Ok Cancel

Select a Type

Target Ontology

- j.0:CpuType
  - j.0:AMD
  - j.0:SUN
  - j.0:Motorolla
  - j.0:Intel
    - j.0:Itanium
    - j.0:Pentium

Resource Monitor (No. 1)

Sample Chart

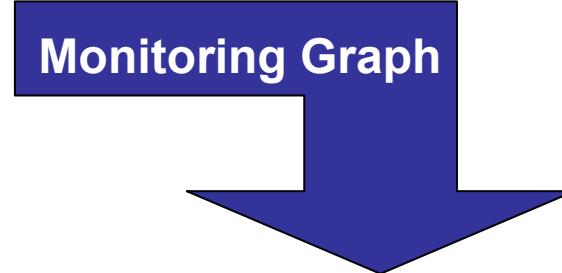
Net Processor Load (ms/1000ms)

Time

Legend:  
Avg. + StdDev. / 2  
Avg. + StdDev.  
Avg.  
Avg. - StdDev.  
Avg. - 2\*StdDev.  
Val.

Data Table:

Id	ratProcessorLoadBase7d	ratProcessorLoadForProcessorLoadLast1MTimeAve	avgLast1Min	stdDevLast1Min
1	Windows server-0-0 load	17	12.22803170343933	11.78401828282462



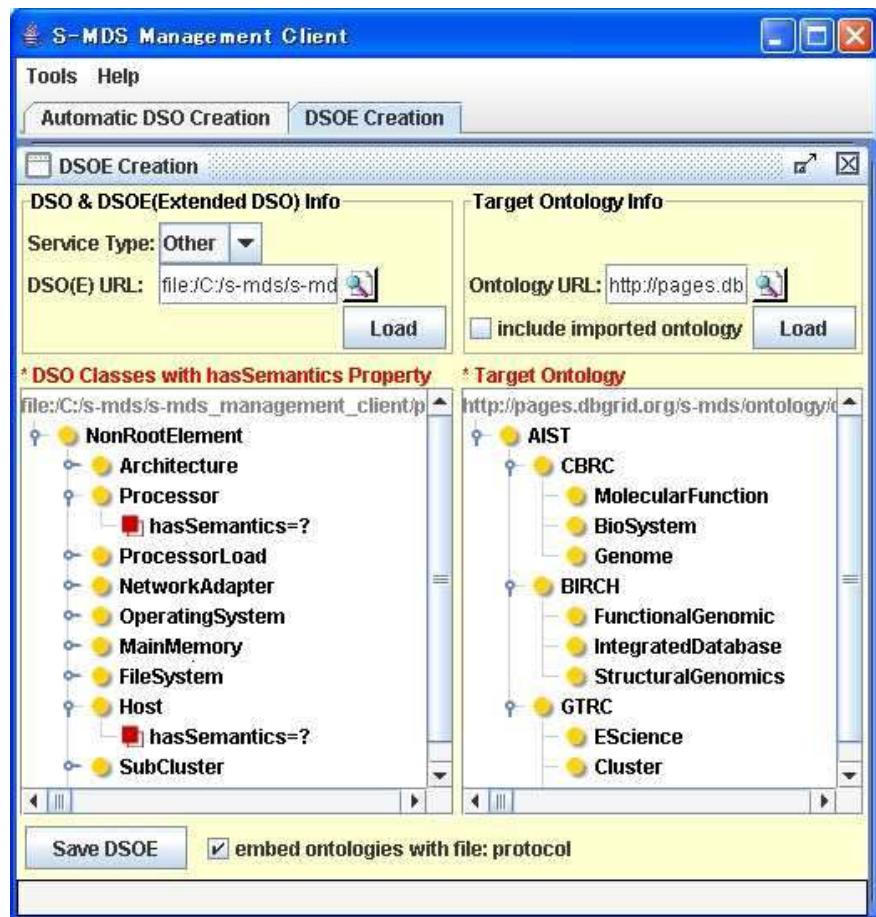
# SPARQL Query (generated by our GUI)

```
SELECT distinct ?ns24ProcessorLoadHasProcessorLoadLast1MinVal
WHERE {
    ?ns24Host ns24:hasProcessor ?ns24Processor.
    ?ns24Processor ns24:hasSemantics ?ns24ProcessorHasSemantics.
    ?ns24ProcessorHasSemantics rdf:type j.0:Pentium.
    ?ns24SubCluster ns24:hasHost ?ns24Host.
    ?ns24ProcessorLoad ns24:hasProcessorLoadLast1Min
        ?ns24ProcessorLoadHasProcessorLoadLast1Min.
    ?ns24Host rdf:type <http://www.dbgrid.org/namespaces/s-mds#Host>.
    ?ns24ProcessorLoadHasProcessorLoadLast1Min ns24:value
        ?ns24ProcessorLoadHasProcessorLoadLast1MinVal.
    ?ns24Cluster rdf:type <http://www.dbgrid.org/namespaces/s-mds#Cluster>.
    ?ns24Host ns24:hasProcessorLoad ?ns24ProcessorLoad.
    ?ns24Processor rdf:type <http://www.dbgrid.org/namespaces/s-mds#Processor>.
    ?ns24Cluster ns24:hasSubCluster ?ns24SubCluster.
    ?ns24SubCluster rdf:type <http://www.dbgrid.org/namespaces/s-mds#SubCluster>.
    ?ns24GLUECE rdf:type <http://www.dbgrid.org/namespaces/s-mds#GLUECE>.
    ?ns24Host ns24:hasSemantics ?ns24HostHasSemantics.
    ?ns24HostHasSemantics rdf:type j.1:DataGrid.
    ?ns24GLUECE ns24:hasCluster ?ns24Cluster.
    ?ns24ProcessorLoad rdf:type <http://www.dbgrid.org/namespaces/s-mds#ProcessorLoad>
}
```

# Demo of S-MDS GUI (1)

- Resource Discovery
- Resource Monitoring

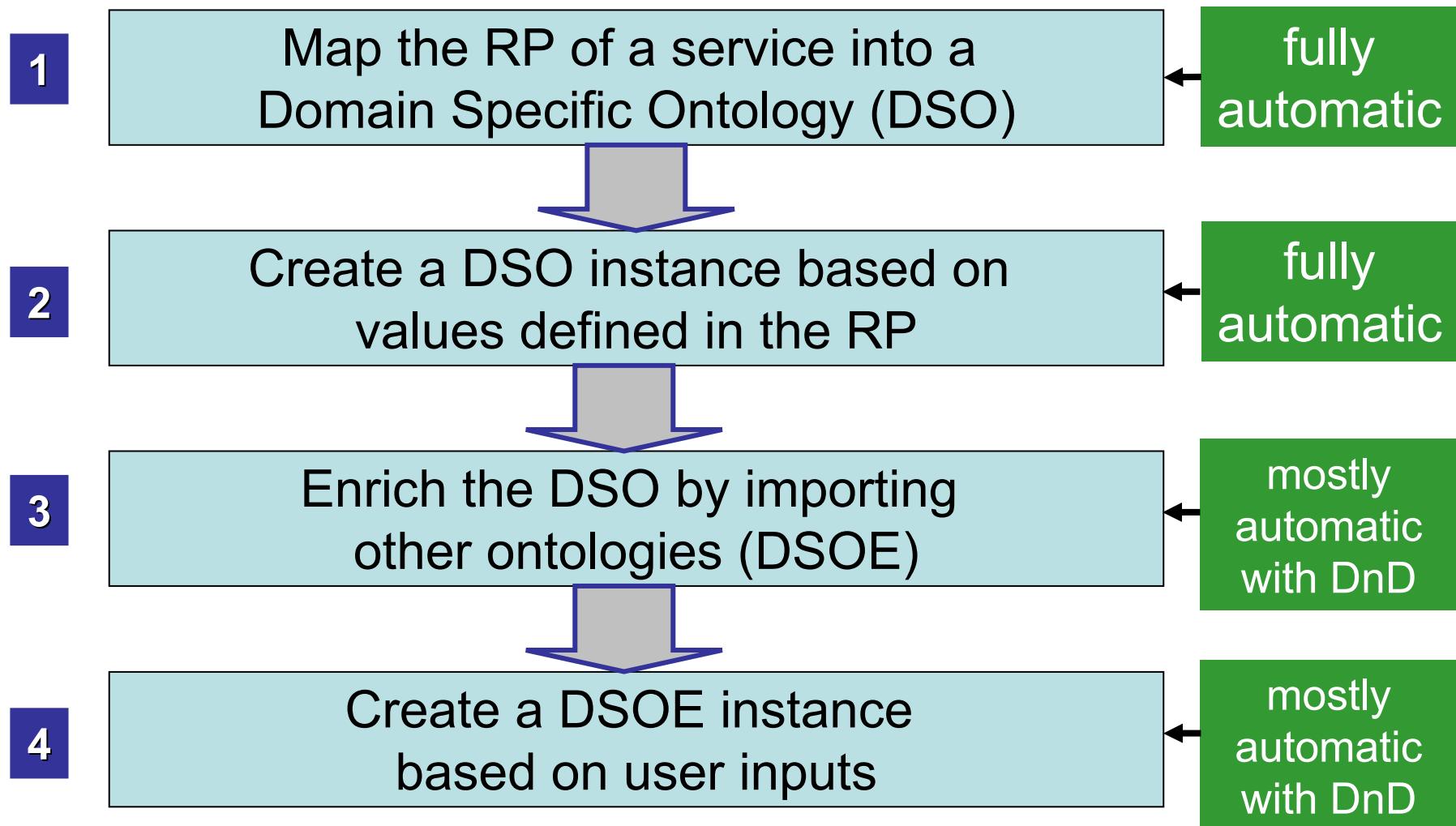
# GUI-based Tool for Ontology Mapping



## Setup procedure with GUI

1. Map the ResourceProperty to OWL-S ontology (Domain Specific Ontology, DSO).
2. Create Instances for Target Resources (DSO Instances).
3. Add a user-defined ontology to DSO (DSO enriched, DSOE).
4. Create Instances with DSOE (DSOE Instances).

# Steps for Ontology Mapping



# Automatic DSO Creation (Create a Class)

Domain Specific Ontology (OWL-S)

ResourceProperties (XML)

* DSO Classes with Mapping Info	
○	● GLUECE
○	● NonRootElement
○	■ element=?
○	■ uniqueAtt=?
○	● Architecture
○	● Processor
○	● Policy
○	● ProcessorLoad
○	● NetworkAdapter
○	● OperatingSystem
○	● Info
○	● MainMemory
○	● FileSystem
○	● Host
○	■ element=//ns1:GLUECE/ns1:Cluster/
○	■ uniqueAtt=?
○	■ hasHostUniqueId=@ns1:UniqueId
○	■ hasHostName=@ns1:Name
○	● SubCluster
○	● Cluster
○	● State
○	● ComputingElement

Fully  
Automatic  
Mapping

* RP(Resource Properties)	
○	● ns1:condorOS
○	● ns1:GLUECE
○	● ns1:Cluster
○	■ ns1:Name=DBGrid
○	■ ns1:UniqueId=DBGrid
○	● ns1:SubCluster
○	■ ns1:Name=main
○	■ ns1:UniqueId=main
○	● ns1:Host
○	■ ns1:Name=gt3.hpcc.jp
○	■ ns1:UniqueId=gt3.hpcc.jp
○	● ns1:Processor
○	● ns1:MainMemory
○	● ns1:OperatingSystem
○	● ns1:Architecture
○	● ns1:FileSystem
○	● ns1:NetworkAdapter
○	● ns1:ProcessorLoad
○	● ns1:Host
○	● ns1:Host
○	● ns1:ComputingElement

# DSO Instance Creation

## Create Instances of the resources

\* DSO Classes with Mapping Info

- GLUECE
- NonRootElement
  - element=?
  - uniqueAtt=?
- Architecture
- Processor
- Policy
- ProcessorLoad
- NetworkAdapter
- OperatingSystem
- Info
- MainMemory
- FileSystem
- Host
  - element=//ns1:GLUECE/ns1:Cluster/
  - uniqueAtt=?
  - hasHostUniqueId=@ns1:UniqueId
  - hasHostName=@ns1:Name
- SubCluster
- Cluster
- State
- ComputingElement

\* RP(Resource Properties)

- ns1:condorOS
- ns1:GLUECE
- ns1:Cluster
  - ns1:Name=DBGrid
  - ns1:UniqueId=DBGrid
- ns1:SubCluster
  - ns1:Name=main
  - ns1:UniqueId=main
- ns1:Host
  - ns1:Name=gt3.hpcc.jp
  - ns1:UniqueId=gt3.hpcc.jp
- ns1:Processor
- ns1:MainMemory
- ns1:OperatingSystem
- ns1:Architecture
- ns1:FileSystem
- ns1:NetworkAdapter
- ns1:ProcessorLoad
- ns1:Host
- ns1:Host
- ns1:ComputingElement

**There are 3 hosts;  
We must specify the  
unique ID of the host**

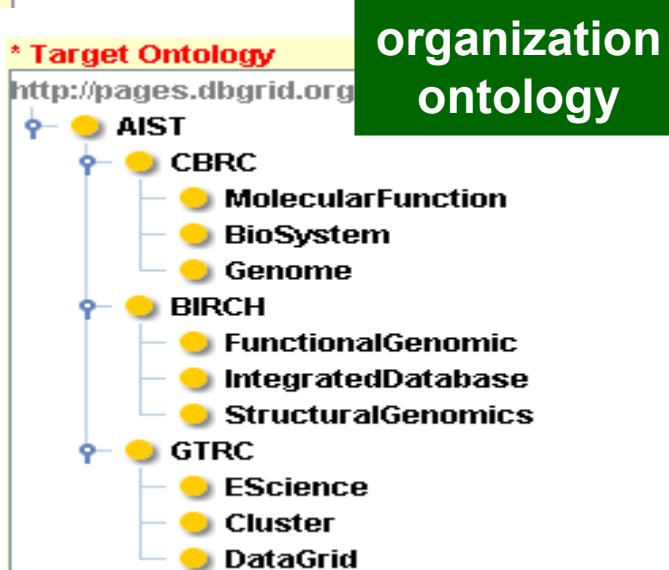
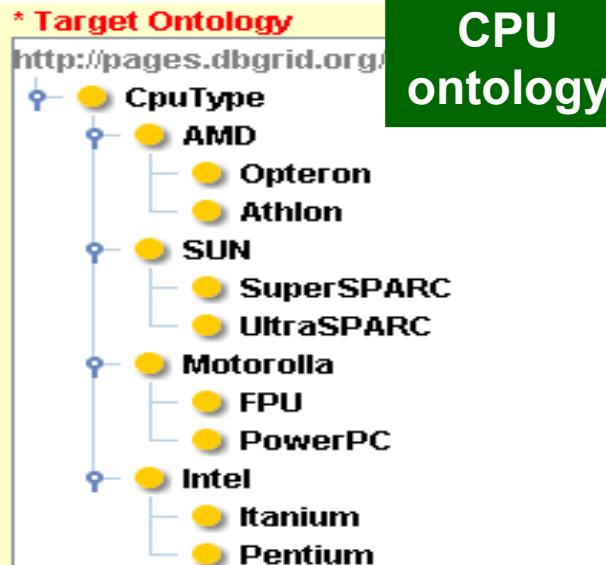
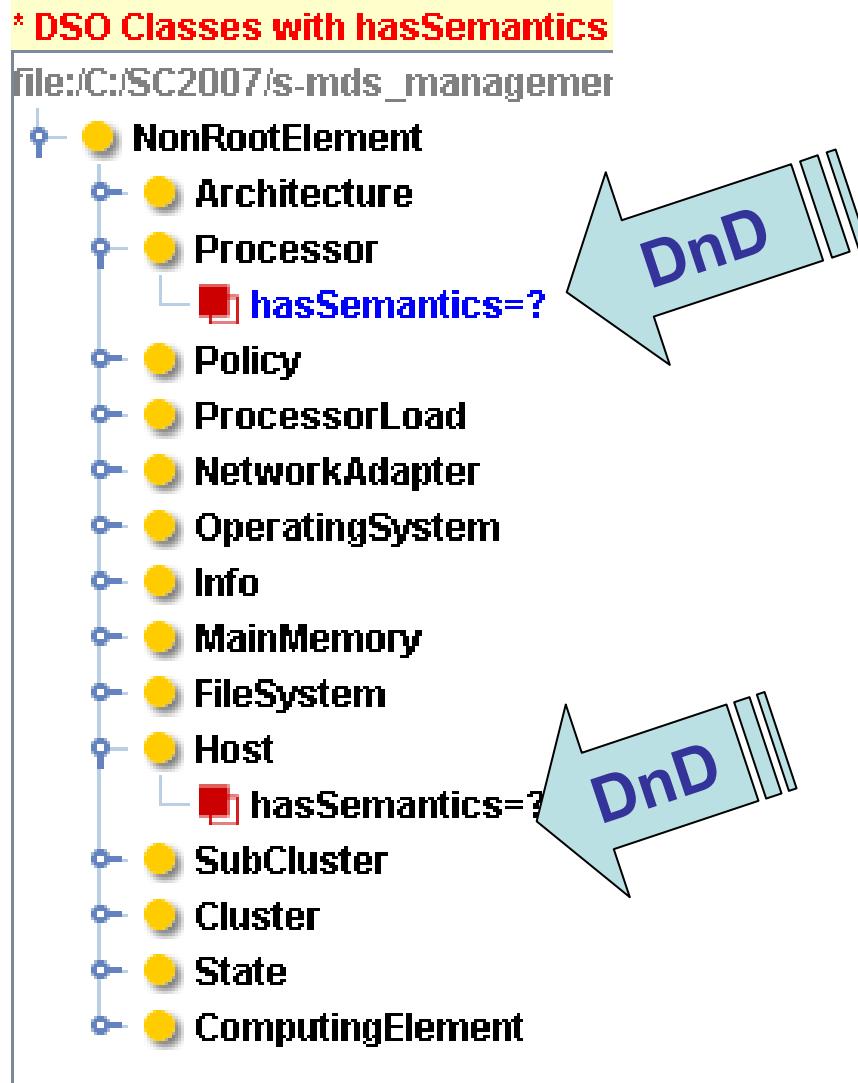
**Drag and Drop**

**Default Value Settings  
= Useful for large clusters**

3

# DSO Enriching/Extending

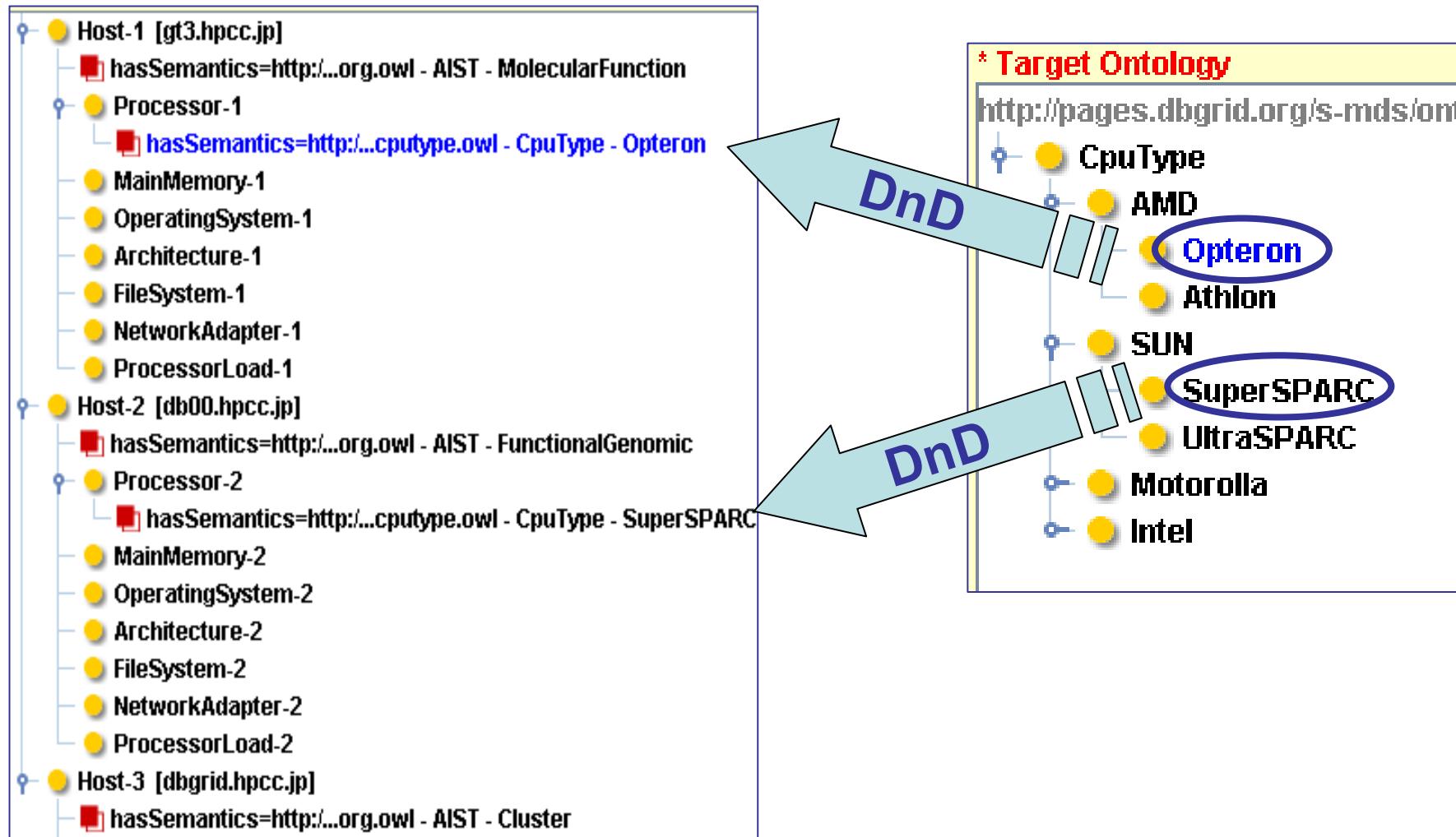
## Associate with other user-defined ontologies



User-Defined Ontologies  
(Created with other Ontology editor)

## 4

# DSOE Instance Creation



# Demo of S-DMS GUI (2)

- Ontology Mapping

# Software Status

- Technical Preview 1 is available at <http://dbgrid.org/smads/>.
  - Please download and install if you're interested.
  - Any feedbacks are welcome.
- Technical Preview 2 will be available soon.
  - Use of UsefulRP interface
  - Functional Enhancements
- Plan to apply Globus Incubator Project

# New Functionalities for Tech. Preview 2

## Rule-based Monitoring

- Record the history of the monitored value into the SMR database
  - LoadAvg, RAM/DiskFree, etc.
  - Ex: Store the history of LoadAvg for last 2 weeks.
- Statistical processing over the historical data
  - Average, Min, Max,,,
  - Ex: Calculate the average and standard deviation of LoadAvg for every updates.
- Rule based Status Description & Detection
  - Based on Jena rule processing module
  - Ex: If LoadAvg is more than  $3 \times \text{stddev} + \text{average}$  of Loadval, then report anomaly.

# Summary

- Introduction of S-MDS
  - Ontology-based information store & retrieval
  - Enhancement of retrieved-information analysis
    - Monitoring specific information by continuous query
    - Rule-based monitoring
- Setting up/Using S-MDS
  - GUI-based management & query tools (You saw the demos.)
- More information about S-MDS
  - <http://dbgrid.org/smds/>
    - Download source code
    - See papers and tutorial
  - If you have any questions, please email to [dbgrid@m.aist.go.jp](mailto:dbgrid@m.aist.go.jp).