Performance Analysis, Data Sharing, Tools Integration: New Approach based on Ontology

Hong-Linh Truong

Institute for Software Science,
University of Vienna, Austria

truong@par.univie.ac.at

Thomas Fahringer

Institute for Computer Science,
University of Innsbruck, Austria

Thomas.Fahringer@uibk.ac.at

http://www.par.univie.ac.at/~truong/projects/perfonto

Outline

- Motivation
- Our Approach
- Ontology & Ontology Languages
- PERFONTO
- Architecture of ontology-based performance analysis, data sharing and tools integration
- Prototype overview
- Conclusions and Future work

Motivation

- Perfomance data and tools integration
  - Performance data mostly is proprietary to a performance tool
  - →Not to be shared and understood by various tools/services
  - How do we share and reuse performance data?
    - Developing wrapper libraries for converting data
    - Making data available in relational database, XML (even just a few did!)
  - →Data integration based on structural approach, not semantic approach
- Performance monitoring tools collect raw data but do not directly model and clarify the relevant aspects of objects monitored
- Grid performance monitoring and analysis
  - No single source provides monitoring and performance data
  - Performance monitoring and analysis are conducted across multiple Grid sites
  - High level services (e.g. scheduler, resource matching) have to utilize data from many sources
- A world wide of performance monitoring tools and data
  - Diversity and autonomy of performance and monitoring tools
  - Syntactic and semantic heterogeneity

Motivation (const.)

- Performance monitoring and analysis resources and applications in Virtual Organizations
  - Both syntactic and semantic interoperability are required
  - Well-defined service/tool operations are needed
- We believe adding more semantics into data collected will
  - → Make data less propriatery to a specific tool, enhancing knowledge sharing and reuse
  - → Foster to automatically detect, correct and predict behavior of systems and applications at runtime (e.g. self-organizing)
  - → Allow to move analysis components as close to monitored source as possible
  - → Support intelligence performance analysis

Our Approach

- Well-defined service/tool operations achieved by employing Grid/Web Services
- Rich semantics and shared vocabularies achieved by using ontology

→We advocate Semantic Grid

- But what is Ontology?
  - Most widely-accepted and -used definition

  "Formal, explicit specification of a shared conceptualization"

  Gruber, 1993

- Ontology in Semantic Web
  - Computer-usable definitions of basic concepts in a domain and the relationships among them.

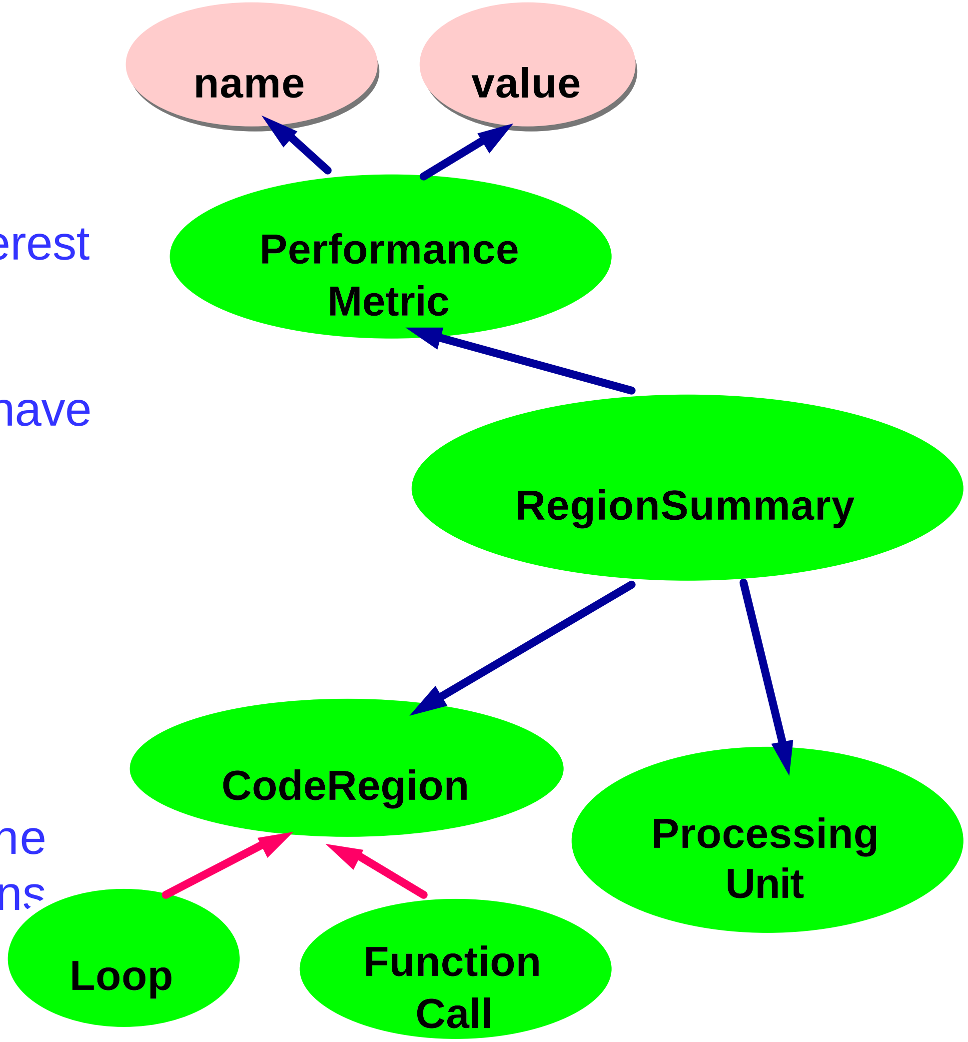
Ontology Primitives

Definitions

- Classes (Things)
  - General things in a domain of interest
- Properties (Attributes)
  - The properties those things may have
- Relationships
  - The relationships that can exist among things

Axioms

- Constrain the interpretation and the well-formed use of these definitions



Expected Benefits of Using Ontology in Performance Analysis Domain

- Performance data and tools integration
  - Describe and model data with rich semantics, highly expressiveness
  - Provide shared vocabularies
  - Map and translate data of disparate representations
  - Semantic interactions between tools in an automatic performance analysis systems
- Enhance automatic performance analysis
  - High level performance data query
  - Enable rule-based performance analysis

Ontology Languages

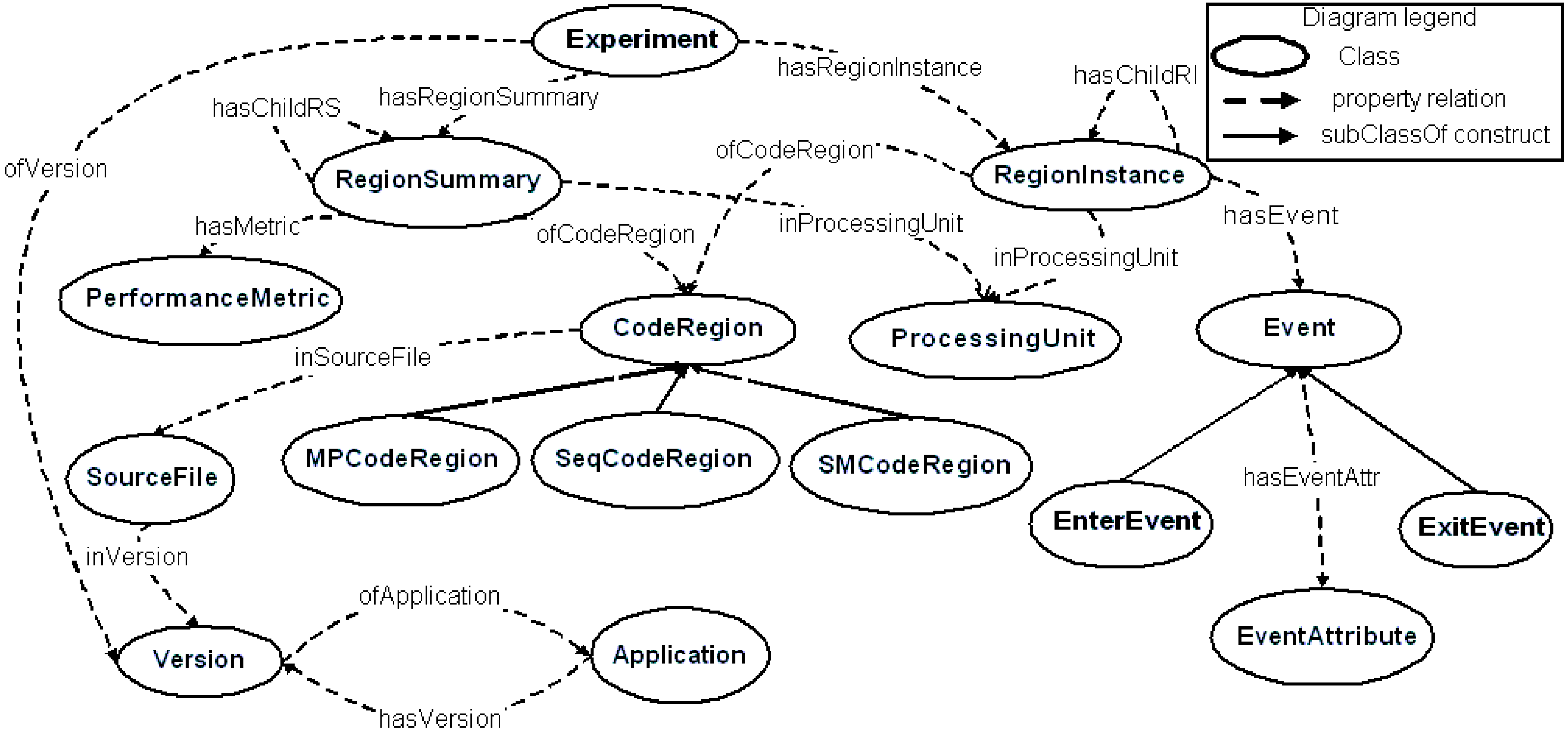
- Existing Languages
  - RDF (Resource Description Framework), DAML (DARPA Agent Markup Language), OIL (Ontology Inference Layer), DAML+OIL, OWL (WebOnt)
- OWL (Web Ontology Language): being standardized by W3C
  - Extension of RDF and derived from DAML+OIL
  - Object-oriented approach
    - Set of definitions of classes and properties
    - Properties: object properties and data properties
      - Transitive, symmetric, inverse, functional
  - Use XML schema data types
  - Class axiom: specifies necessary and/or sufficient characteristics of a class (e.g. sub class, equivalent) ∞
  - Property axiom: defines additional characteristics of a property, e.g. range, domain, relations to other properties

PERFONTO

- Currently initial work on using ontology for system monitoring and management, but not for performance experiments of applications so far
- PERFONTO (ontology for performance analysis domain)
  - Domain of interest: performance monitoring, analysis and data integration in parallel and distributed computing.
  - Using OWL (Web Ontology Language)
  - Model and represent performance data
    - both sytem resources and application experiments
  - Provide shared vocabularies on performance analysis domain
- PERFONTO comprises two parts
  - Experiment-related concept
  - Resource-related concept

Experiment-related concept

Describe application experiments and their associated performance data



Excerp OWL for RegionSummary

Property definition

Class definition

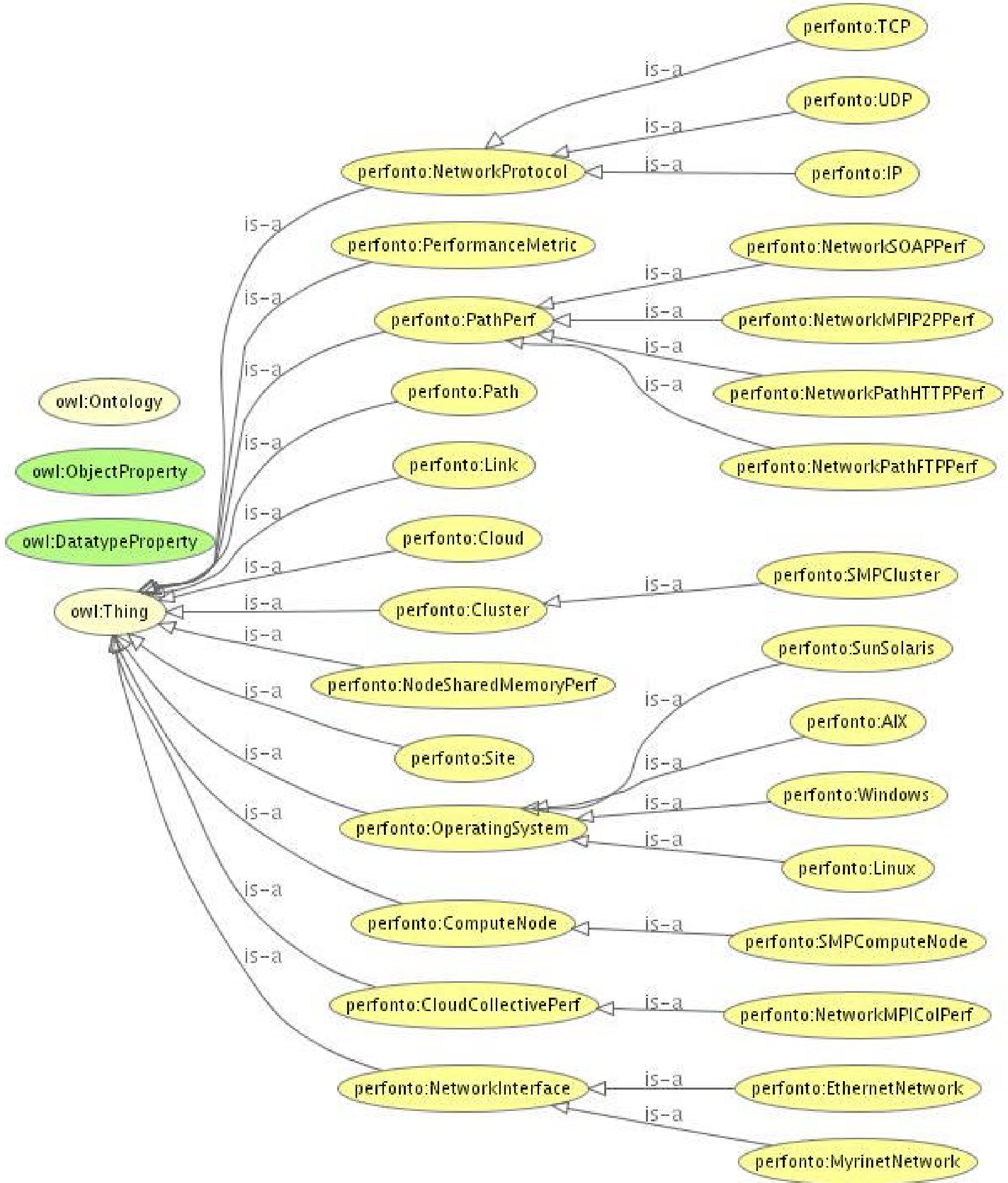
```
- <owl:Class rdf:ID="RegionSummary">
    <rdfs:label>region summary</rdfs:label>
    <rdfs:comment>The concept of a region summary
  </owl:Class>
- <owl:ObjectProperty rdf:ID="inProcessingUnit">
    <rdfs:label>processing unit property</rdfs:label>
    <rdfs:comment>Relationship between Regionsumı
  - <rdfs:domain>
    - <owl:Class>
      - <owl:unionOf rdf:parseType="Collection">
          <owl:Class rdf:about="#RegionSummary" />
          <owl:Class rdf:about="#RegionInstance" />
        </owl:unionOf>
      </owl:Class>
    </rdfs:domain>
    <rdfs:range rdf:resource="#ProcessingUnit" />
  </owl:ObjectProperty>
```

```
- <owl:ObjectProperty rdf:ID="ofCodeRegion">
    <rdfs:label>code region property</rdfs:label>
    <rdfs:comment>Relationship between RegionSummary
    <rdfs:range rdf:resource="#CodeRegion" />
  - <rdfs:domain>
    - <owl:Class>
      - <owl:unionOf rdf:parseType="Collection">
          <owl:Class rdf:about="#RegionSummary" />
          <owl:Class rdf:about="#RegionInstance" />
        </owl:unionOf>
      </owl:Class>
    </rdfs:domain>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="hasChildRS">
    <rdfs:label>sub regionsummary property</rdfs:label>
    <rdfs:comment>Each region summary may have a set
    <rdfs:domain rdf:resource="#RegionSummary" />
    <rdfs:range rdf:resource="#RegionSummary" />
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="hasMetric">
    <rdfs:label>performance metric propety</rdfs:label>
    <rdfs:comment>A region summary contains a set of pe
    <rdfs:range rdf:resource="#PerformanceMetric" />
    <rdfs:domain rdf:resource="#RegionSummary" />
  </owl:ObjectProperty>
```

Resource-Related Concept

- ◆ Describe static, benchmarked and dynamic (performance) information of computing and network systems

- ◆Current descriptions: capabilities static and benchmarked information



Shared Vocabularies/Terminologies

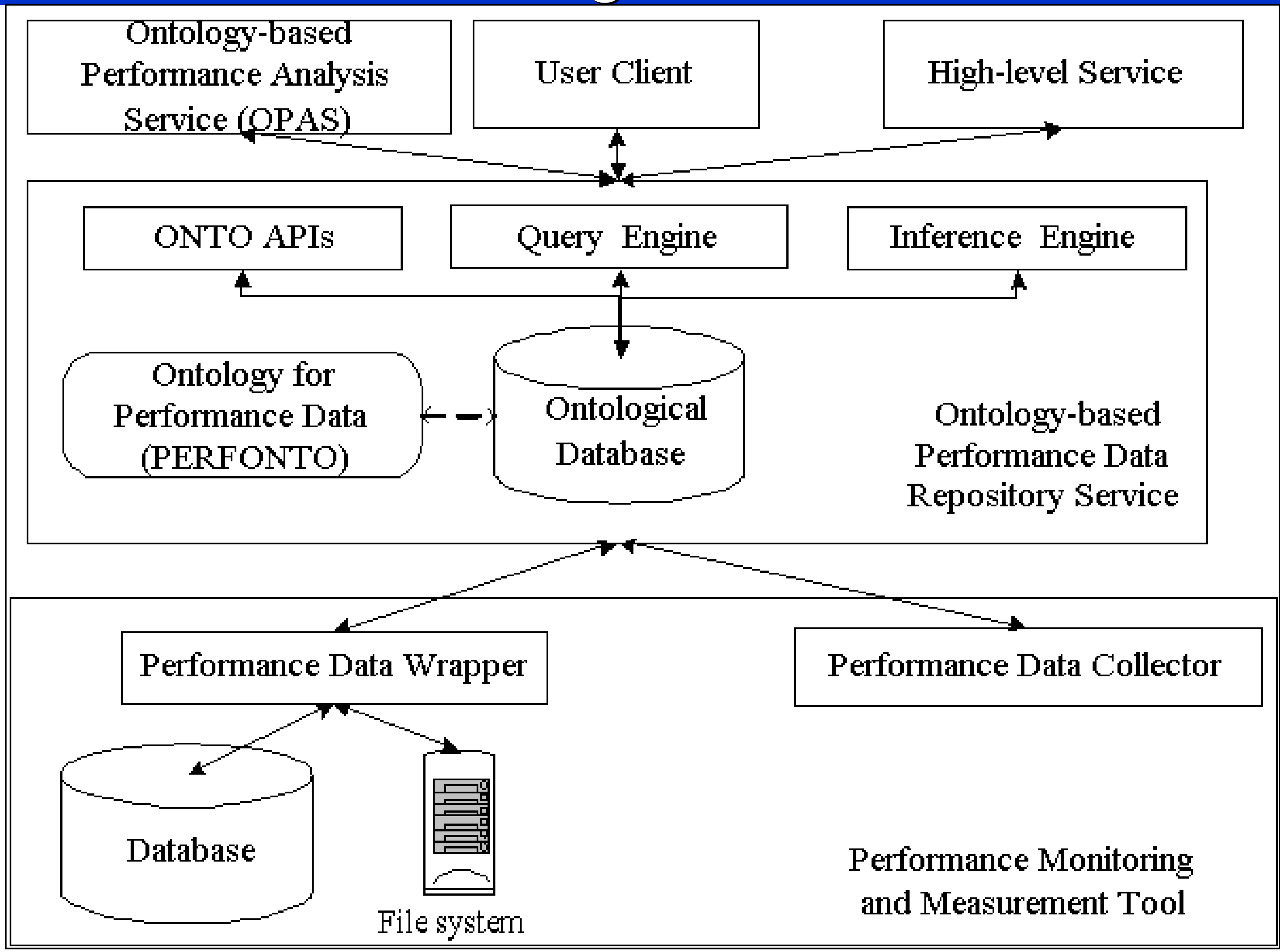
- Performance metric catalog

```
- <perfonto:PerformanceMetricName rdf:ID="FLOPS">
    <rdfs:comment>Floating point instructions per second</rdfs:comment
    <rdfs:isDefinedBy rdf:resource="http://www.par.univie.ac.at/project
  </perfonto:PerformanceMetricName>
- <perfonto:PerformanceMetricName rdf:ID="RES_STL">
    <rdfs:comment>Cycles stalled on any resource</rdfs:comment>
    <rdfs:isDefinedBy rdf:resource="http://www.par.univie.ac.at/project
  </perfonto:PerformanceMetricName>
- <perfonto:PerformanceMetricName rdf:ID="FP_STAL">
    <rdfs:comment>Cycles the FP unit(s) are stalled</rdfs:comment>
    <rdfs:isDefinedBy rdf:resource="http://www.par.univie.ac.at/project
  </perfonto:PerformanceMetricName>
- <perfonto:PerformanceMetricName rdf:ID="TOT_CYC">
    <rdfs:comment>Total cycles</rdfs:comment>
    <rdfs:isDefinedBy rdf:resource="http://www.par.univie.ac.at/project
  </perfonto:PerformanceMetricName>
- <perfonto:PerformanceMetricName rdf:ID="IPS">
    <rdfs:comment>Instructions per second</rdfs:comment>
    <rdfs:isDefinedBy rdf:resource="http://www.par.univie.ac.at/project
  </perfonto:PerformanceMetricName>
```

- Example: similarity in meanings of vocabularies

```
<papi:metric rdf:ID="PAPI_FLOPS">
        <owl:sameAs rdf:resource="perfonto:FLOPS" />
</papi:metric>
```

System Architecture for Ontology-based Performance Analysis, Data Sharing and Tools Integration



Prototype Implementation

- Ontology-related tasks done by Jena Toolkit
  - A Semantic Web Toolkit, HP Lab at UK
  - APIs for processing RDF, RDFS, OWL
  - Network APIs for accessing remove RDF database
  - Data storage: file or persistent database (e.g. PostgreSQL)
  - Query with RDQL (RDF Query Data Language)
  - Rule-based inference engine

Prototype Implementation (const)

- Ontology-based performance data repository service
  - Grid serivices based on GT 3.2 (Globus Tookit)
  - Persistent data storage powered by PostgreSQL
  - Store, retrieve and query operations for ontology and instances (individuals)
  - Reasoning done at client side
- Ontology-based performance analysis service(OPAS)
  - First support search on performance data
  - Validating performance and monitoring data collected with PERFONTO
  - Work on rule-based performance analysis

Search on Ontological Data

- Ontology-based performance data search
  - High-level, more intelligent search model based on ontological data
  - Query easily understood and written by end users, not only by tool developers
- RDQL (RDF Data Query Language)
  - SQL-alike query language for RDF
    - SELECT vars, FROM documents, WHERE expressions, AND filters, USING namespace declarations
  - Query with triple patterns and constraints over RDF model
  - Implemented in Jena search engine

Search on Ontological Data (const.)

Q: *Search all code regions executed in node "gsr410" with wallclock time >=3E8 microsecond*

```
SELECT ?regionsummary
WHERE
   (?regionsummary   perfonto:inProcessingUnit   ?processingunit)
   (?processingunit  perfonto:inNode             "gsr410")
   (?regionsummary   perfonto:hasMetric          ?metric)
   (?metric          perfonto:hasMetricName      "wtime")
   (?metric          perfonto:hasMetricValue     ?value)
AND (?value >=3E8)
USING perfonto FOR
<http://www.par.univie.ac.at/project/scalea/perfonto#>
```

Reasoning about Ontological Data

- Ontology allows additional facts to be inferred by using axioms and rules
- Reasoning on monitoring and performance ontological data
  - Intelligence resource matching
  - Rule-based interferences
  - Validating data
- Automatic/autonomic monitoring and performance analysis
  - Automatic performance analysis based on rules
  - Runtime self-healing, -management

Reasoning Example

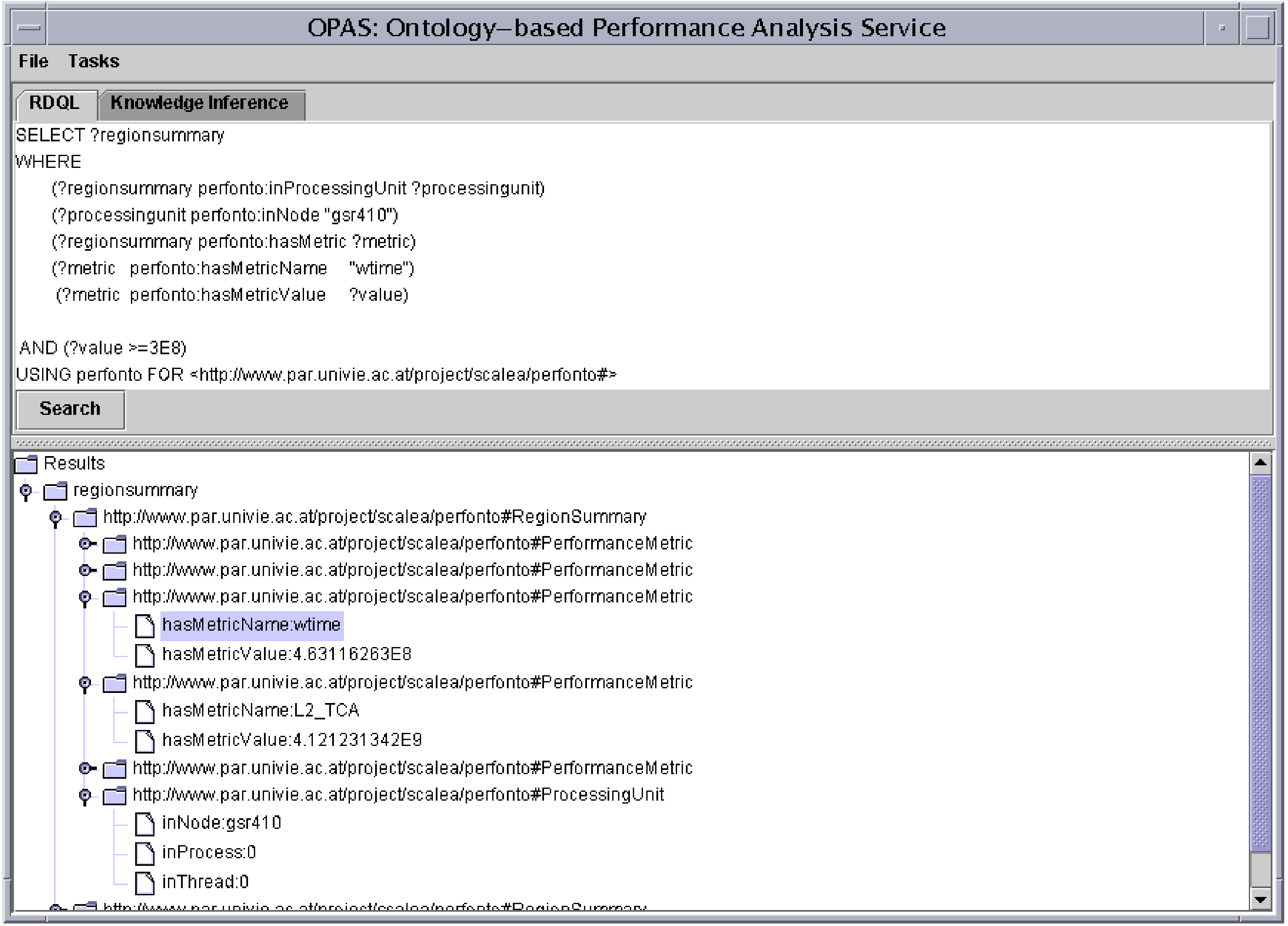
Q: if a MPI *code region has big message size then prints it out*

```
[rule_detect_bigmessages:
(?regionsummary     perfonto:ofCodeRegion          ?codeRegion),
(?codeRegion        rdf:type            perfonto:MPCodeRegion),
(?codeRegion        perfonto:hasCrType     "CR_MPIP2P"),
(?regionsummary     perfonto:hasMetric     ?metric),
(?metric            perfonto:hasMetricName "AvgMessageLength"),
(?metric            perfonto:hasMetricValue        ?length),
greaterThan(?length, BIG_MESSAGES_THREADHOLD)

->

print(?regionsummary,"Big message hold!")]
```

Search OPAS GUI: Example



Conclusion and Future work

- Conclusion
  - Investigating the application of ontology to the domain of Grid performance analysis and monitoring
  - System architecture for ontology-based performance data analysis, data sharing and tools integration in the Grid
  - → Ontology would be a good solution for seamlessly utilizing monitoring and performance data and integrating different performance monitoring and measurement tools in Grids
  - → Ontology could help to simplify performance analysis (e.g. performance data search) and to enhance automatic performance analysis (e.g. rule-based reasoning on performance data)
- Future work
  - Implementing full prototype, reevaluating and enhancing PERFONTO
  - Ontology-based monitoring
  - Task-based ontology for automatic performance analysis

Shared conceptualization → looking for community work!

http://www.par.univie.ac.at/~truong/projects/perfonto