Metadata in a Data Grid Construction

Krzysztof Kaczmarski*, Piotr Habela†, Kazimierz Subieta‡

*Faculty of Mathematics and Information Science
Warsaw University of Technology
†Polish-Japanese Institute of Information Technology,
‡Institute of Computer Science, Polish Academy of Sciences
Usage of metadata

- Grid architecture, communication paths
- Participants privileges
- Data structures
- Requirements for participating nodes
- Replication descriptions
- Optimization possibilities
- Data semantics
Grid construction realistic methodology

**Strategic phase**

decision on creating a Grid is made (gov. initiative, . . . )

**Analysis phase**

existing resources are elaborated (heterogeneity, incompleteness, . . . )

**Design phase**

precise definition of global schema and contributions of all participants

**Finalization phase**

all participants sign final agreement

**Implementation phase**

all necessary data transformations are implemented, grid is created
Grid Construction – step 1

- Strategic phase
- Analysis phase
Grid Construction – step II

- Design phase
- Finalization phase
Grid modeling schemas

Local schema
  describing node’s internal database

Global schema
  grid database available for grid's users

Contributory schema
  description of node’s shared data

Integration schema
  informal description of integration and semantics
Grid’s modeling language

- description of objects, services (interfaces)
- ability to be cut into parts describing roles for nodes
- ability to describe data attributes required by data integration
- notions for replications’ descriptions
- reuse a global schema as a contributory schema (grid embedding)
Usage of schemas – data transformation

Results of a design phase - constraints for integration
Example

Metadata in a Data Grid Construction
Example
Basic Metamodel
Summary of metamodel features

- The most important extensions
  - single and composite uniqueness keys
  - replication path for database cooperation
- Data structure
  - subobjects (object’s embedding)
  - operations signatures
  - object roles
  - object inheritance
  - extended attribute flags
Metadata in a Data Grid Construction

Extended Metamodel

- **MetaObject**
  - +name

- **Type**
  - +contents 1
  - +sub *
  - +super *

- **Interface**
  - +instanceName
  - +base 1
  - +super *
  - +target 0..1
  - +owner
  - +usage *
  - +referrer *

- **RoleInterface**
  - +applicableRole *

- **Feature**
  - +feature *

- **Operation**
  - +owner
  - +usage *

- **SubobjectLink**
  - +usage *
  - +reverse 0..1

- **AssocLink**
  - +reverse 0..1

- **StructFeature**
  - +multiplicity
  - +isOrdered
  - +isReadable
  - +isMutable
  - +isInsertable
  - +isRemovable

- **PeerDatabase**
  - +location
  - +replicationTo *
  - +replicationFrom *
  - +repFromItem *
  - +repToItem *
  - +keyMembership *
  - +keyElement *

- **Key**
  - +key
  - +uniquenessScope

13
Uniqueness Keys Example

```plaintext
[Diagram of data grid relationships]
```
Metamodel's Instance
Layered Architecture

- Global schema language conforms with contributory schema language
Conclusions

• A methodology for modeling Grid database was presented
  – well defined phases and activities
• A presented Grid metamodel fits this scenario
• Implementations
  – DDL grammar
  – UML-like graphical language (profile)
• Future Work
  – Integration schema – Updateable View modeling language
  – Additional features for database architecture
Thank You.