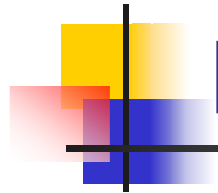# Data Service Semantics

Norman Paton

University of Manchester
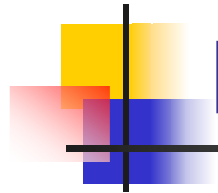
DAIS Core Specification

- Recent principles:
  - Conform to OGSA Data Services Framework.
  - Develop a separate specification for request aggregation.
  - Delegate more complex delivery scenarios to Data Distribution specification.

- Recent impediments:
  - Significant effort devoted to OGSA Data Services (packaging) issues.
  - Technical and political distraction of pending WS-Resource announcements.

Global Issues

- How should the core specification be organised.
- The scope of the functionality provided by the core.
- The semantics of the operations and agreement terms.
- The way the semantics of the specification are described.
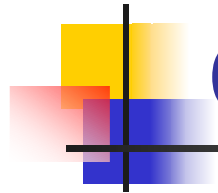
How to Organise Specification

- Currently:
  - Grid Data Service Specification.
  - Relational Realisation.
  - XML Realisation.

- Proposal:
  - Keep with three documents; separate core is open to new realisations.
  - Stick with current OGSI and Data Services for GGF10.

Organising Future Versions

- Principles:
  - Identify all dependencies on not-yet-accepted standards.
  - Write specifications to minimise dependences on not-yet-accepted standards (currently these dependencies permeate specifications).

- Practice:
  - Identify dependencies by GGF10.
  - Discuss proposed document structures at GGF10.
  - Recast specifications to new structures for GGF11.
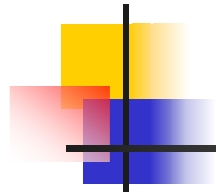
Potential Structure

- Introduction.
- Notation.
- Data Description:
  - Need to say X,Y,Z.
  - Schemas for X,Y,Z.
- Data Access:
  - Need to support A,B.
  - Semantics for A,B.
  - Operations.
- Data Management:
  - Something/nothing?
- Mapping to WSRF (or descendent):
  - Data Description WSDL.
  - Data Access WSDL.
- Security considerations.
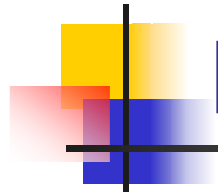- Conclusions.

Scope of Functionality

- Currently:
    - Synchronous requests.
    - Limited asynchronous requests.
    - Significant ongoing work on service data.
    - Recent activity on management.

- Proposal:
    - Request scope broadly as before, but nail semantics as top priority.
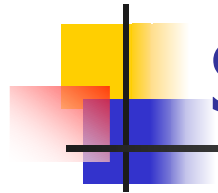   - Strictly categorise service data and management material to make explicit external dependencies.
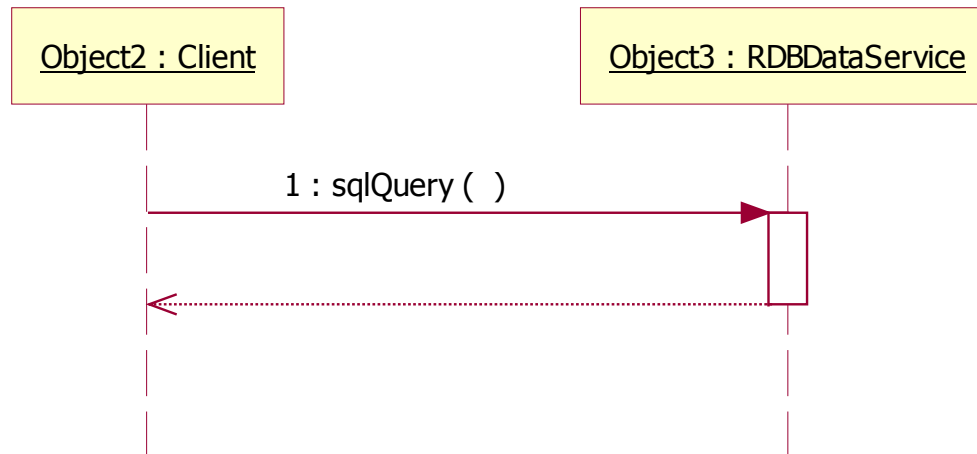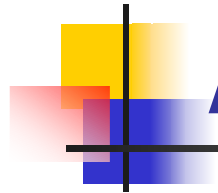
Relating to Data Distribution

- Data distribution helps core by:
  - Supporting flexible data delivery.
  - Push/pull, many third parties, protocols, etc.
- Data distribution is a paradigm shift to publish/subscribe, so some support for asynchronous may be best left in the core.
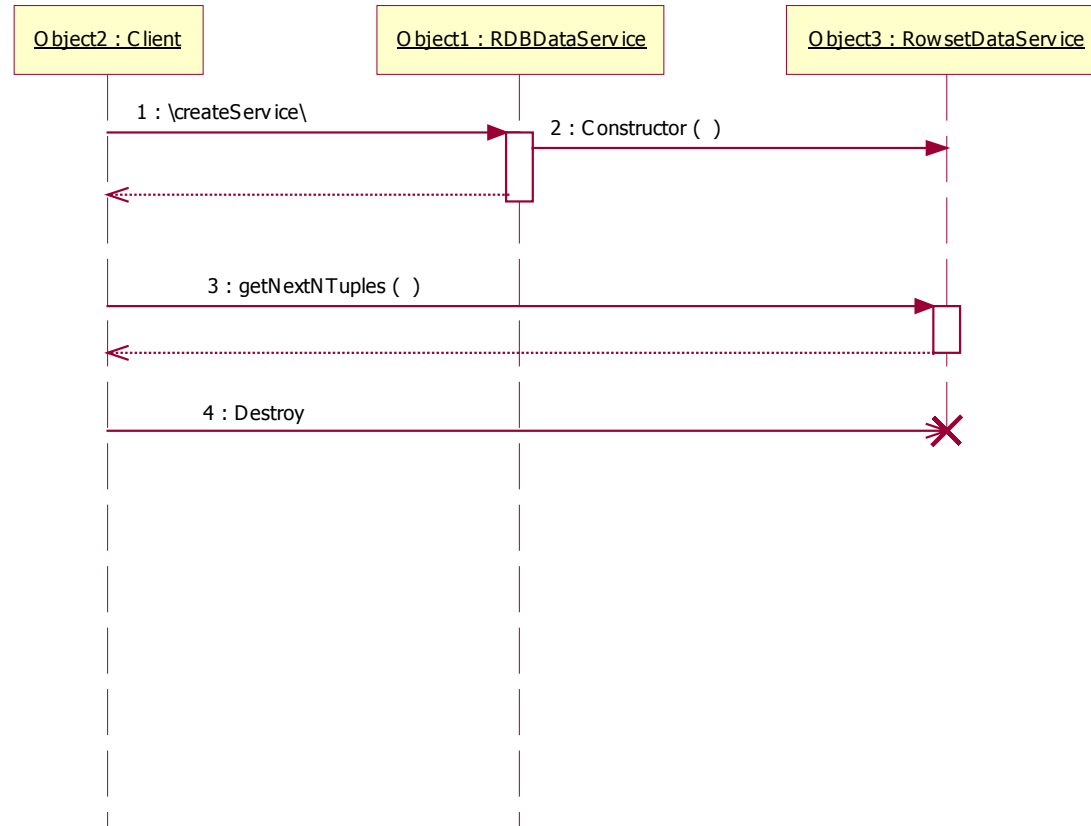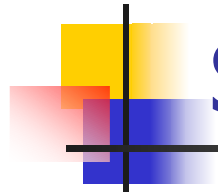
Synchronous Requests

Asynchronous Requests

Object2 : Client	Object1 : RDBDataService	Object3 : RowsetDataService

1 : \createService\

2 : Constructor ( )

3 : getNextNTuples ( )

4 : Destroy

Semantics to be captured

- Synchronous:
  - Transaction model.
  - Sequential consistency.
  - Isolation level.

- Asynchronous:
  - *Direction* [fwd/back].
  - *Sensitivity*.
  - *Concurrency* [r/rw].
  - *Holdability* [over tx boundary].
  - Materialisation.
  - Read Caching.
  - Write Caching.

Issues on semantics

- What terms should be captured?
- How these are expressed in spec?
- How these are defined precisely?
- Who does this for GGF10 (currently SL, MA, NWP)?

- Proposal:
  - Now: discuss the terms to be captured.
  - For GGF10: model semantics in the UML. Aim to finalise shortly afterwards.
  - GGF10 specification: describe semantics informally, with UML in associated slides.