A Service Infrastructure for E-Science: the case of the ARION System

Catherine Houstis, Spyros Lalis, Vassilis Christophides, Dimitris Plexousakis, Manolis Vavalis, Marios Pitikakis, Kyriakos Kritikos, Antonis Smardas, Charalampos Gikas

Institute of Computer Science, Foundation for Research and Technology – Hellas

Workshop on

"Web Services, e-Business, and the Semantic Web: Foundations, Models, Architecture, Engineering and Applications"

In conjunction with CAiSE'02

The Fourteenth International Conference on Advanced Information Systems Engineering

Toronto, Ontario, Canada
27-28 May, 2002

Objective

- Develop an active repository for Scientific Collections
  - Develop ontologies / metadata for scientific domains of knowledge
  - Develop dynamic resource invocation
  - Develop resource export procedures
  - Support data filtering and calibration
  - Develop a lightweight system with minimal administration
  - Exploit basic services for building advanced e-science services

Content : data sets, programs, tools

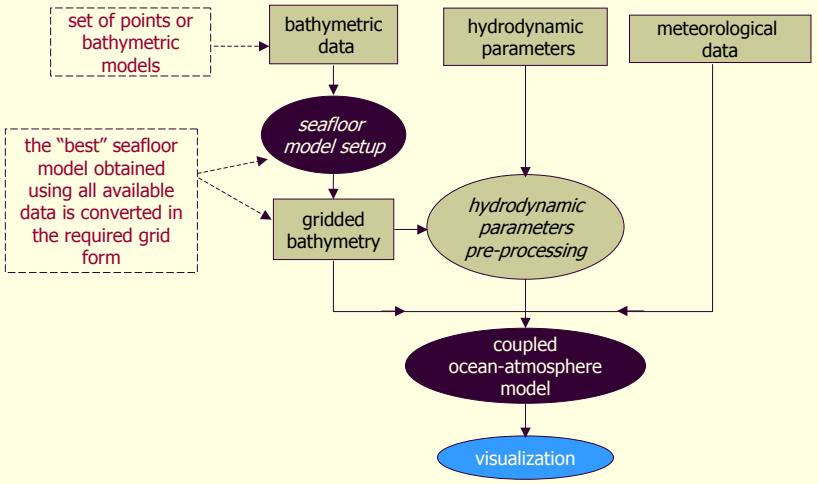
- Data sets in various domains of knowledge
  - measured data (monitored)
  - produced data
- Programs (simulation models of physical phenomena)
- Tools
  - statistical tools
  - visualization tools
  - data filters
- Object collections (applications)
- Digital repositories (organizations, companies, research centers)

A Service Infrastructure for E-Science: the case of the ARION System
http://arion-dl.org

Services

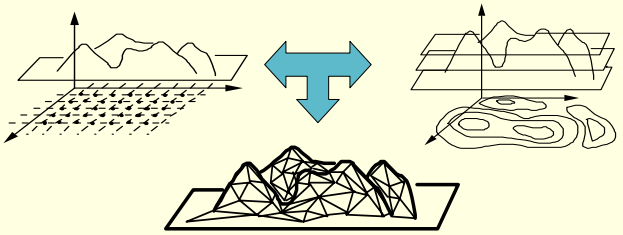
- Search and retrieval via coordinated ontology / ontology-driven metadata queries.
- On-demand online data set production via workflows
- **Motivating example**: *examine chlorophyll data in a particular geographic area.*

Example Scenario 1: Description

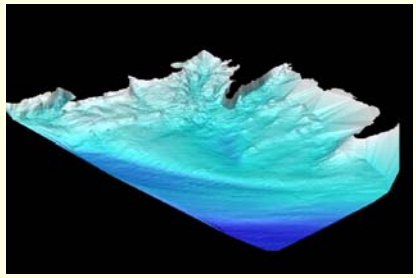


A Service Infrastructure for E-Science: the case of the ARION System
http://arion-dl.org

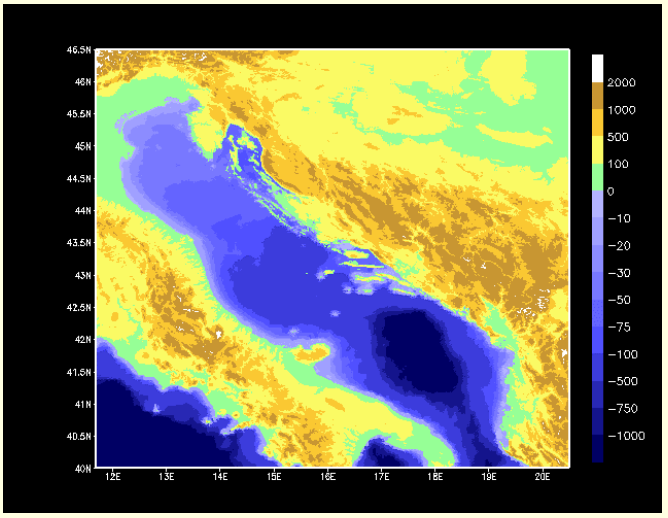
Example Scenario 1: Bathymetric models and Tools



all relevant data are used and merged into a bathymetric model

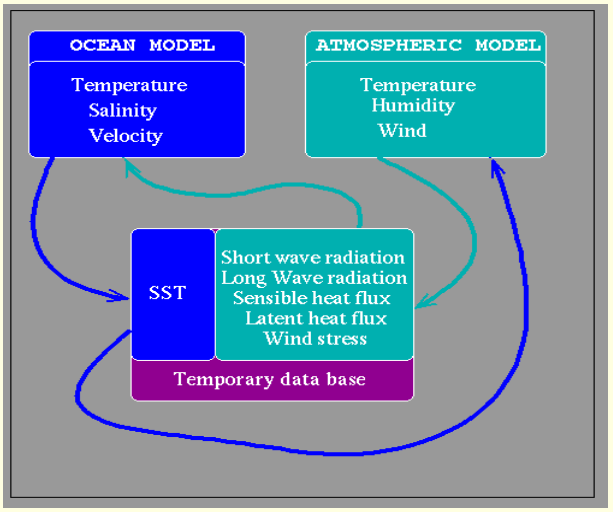


Example Scenario 1: Adriatic Sea

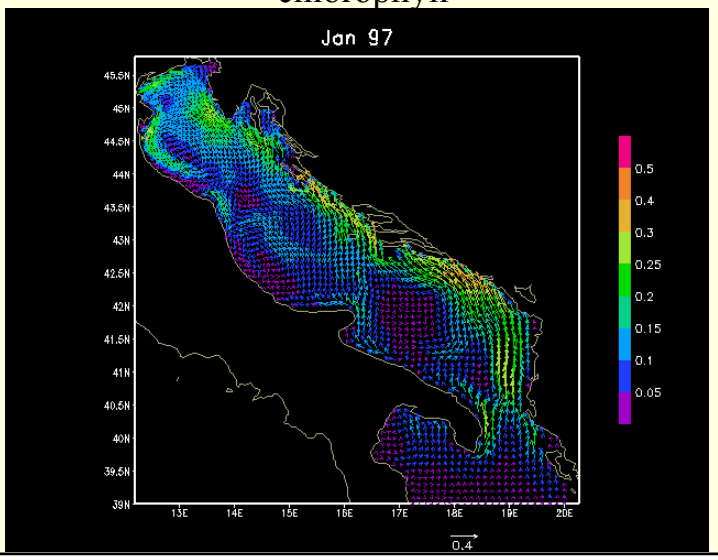


A Service Infrastructure for E-Science: the case of the ARION System
http://arion-dl.org

Example Scenario 1: Coupled model



Example Scenario 1: Climatological monthly chlorophyll

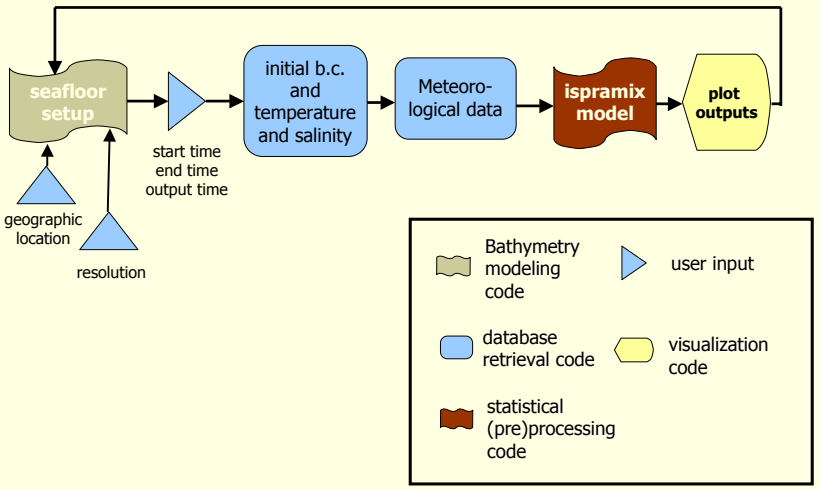


A Service Infrastructure for E-Science: the case of the ARION System
http://arion-dl.org

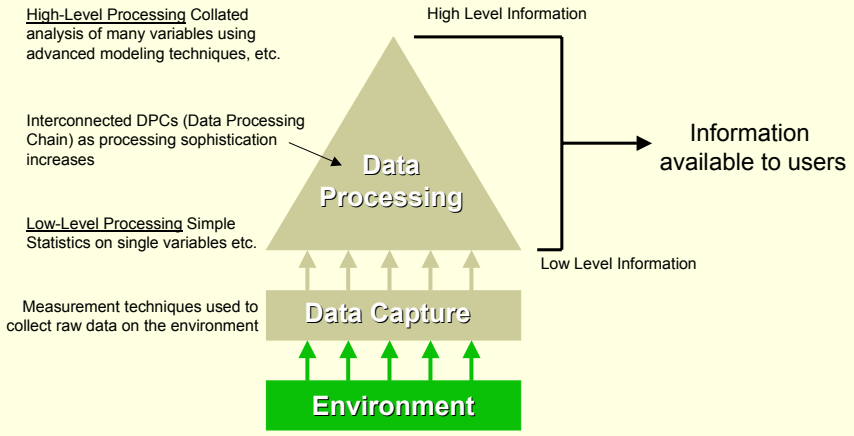
ARION Steps for Scenario 1

1. Search for chlorophyll data
   - for given spatio-temporal parameters
2. If no data sets are found, return a workflow specification to generate the data
3. Specify calibration parameters
4. Run workflow

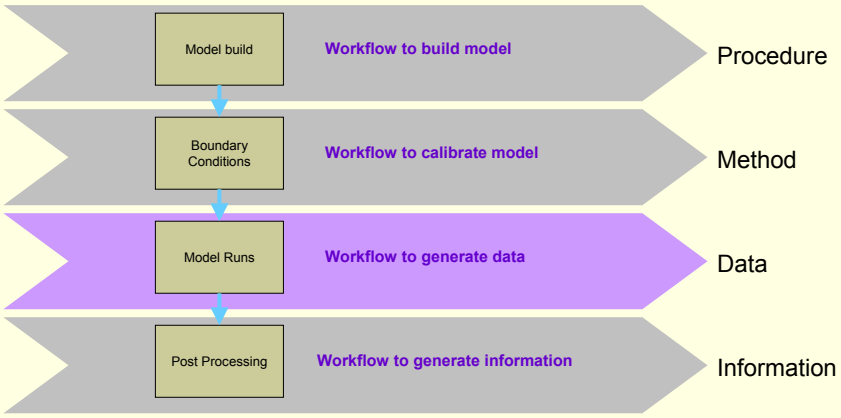
Example Scenario 1: Workflow description



Providing Information

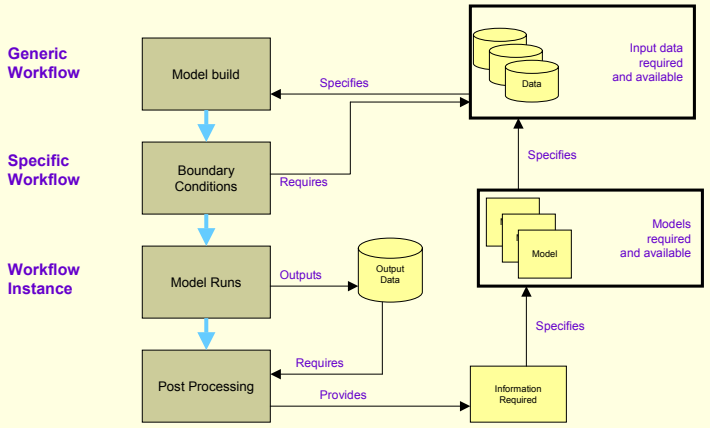


Different workflows for different processes



A Service Infrastructure for E-Science: the case of the ARION System
http://arion-dl.org

Relationships to the models and data used



Contributions

- Describe / Query data and models using ontologies
- Specify scientific workflows across the web
- Relate scientific data / models and workflows
- Execute distributed workflows

A Service Infrastructure for E-Science: the case of the ARION System
http://arion-dl.org

Technologies

- Semantic Web
  - Knowledge Engineering
    - Ontologies
    - RDF Metadata Management Tools (RDFSuite)
- Semantic Grid
  - Knowledge Engineering
    - Workflows
      - Associated suite of tools, XML based
      - Agent runtime system
- Standards: XML, RDF, XRL, Grasshopper (MASIF)

Scientific Collection Semantics

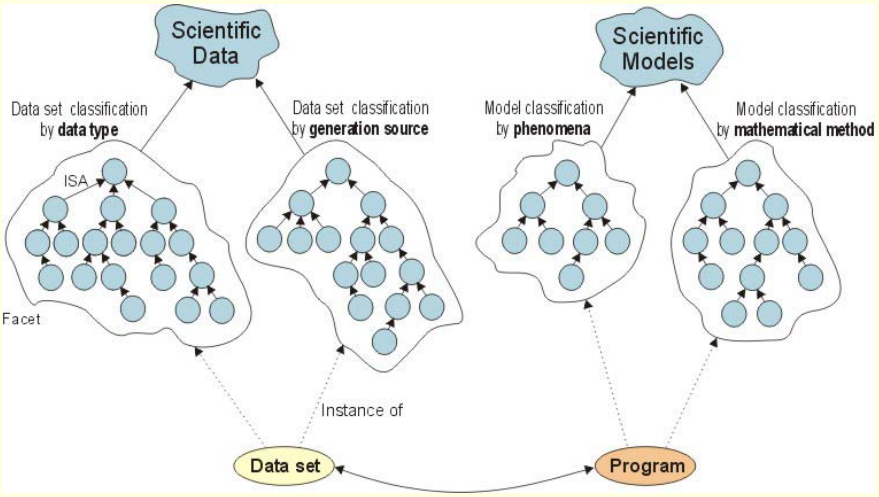
- Digital objects placed within a domain of knowledge are expressed with
  - ontologies
  - workflows
- Ontologies express concepts and their relationships among them.
- Workflows express scientific computations

A Service Infrastructure for E-Science: the case of the ARION System
http://arion-dl.org

Ontology based knowledge representation

- Focus on common semantics for scientific collections.
- Working on an environmental (ocean wave) ontology consisting of a collection of different facets.
- The ontology definition contains an “IS-A” hierarchy of relevant domain concepts, relationships between concepts and properties of concepts.
- RDF and RDF Schema are used as the basis for modeling our domain of knowledge.
  - metadata description is ontology-driven.
  - every object (data set or model) is described by a collection of attributes (properties), inherited from its parent-class or native to the specific object.

Ocean wave ontology



A Service Infrastructure for E-Science: the case of the ARION System
http://arion-dl.org

Conceptual querying

- We use ontologies in four distinct ways:
  - to pose queries for either metadata, concepts or properties of the ontology
  - as a conceptual framework to help the user think about the information repository and formulate queries
  - as a guide to understand the ontology-driven metadata
  - to drive the user interface for creating and refining queries
- We use RDFSuite, a suite of tools for RDF metadata management, providing storage and querying both RDF descriptions and schemas. It consists of three main components: a RDF validating parser (VRP), a RDF schema-specific storage database (RSSDB) and a query language (RQL).

XML-based workflow specifications

- Workflow specifications are generated from our XML-based workflow specification language, which extends XRL to match our model of scientific workflows.
- Language constructs:
  - sequence,
  - parallel-sync (tasks are executed in parallel),
  - conditional if-else
  - while-do
  - can be combined and attached to a workflow definition
  - these constructs are sufficient to model the scientific logic of a workflow.

A Service Infrastructure for E-Science: the case of the ARION System
http://arion-dl.org

Agent Runtime technology

- Workflows ≈ multi-agent cooperation
  - software agents used to perform tasks (computational processes)
  - workflow used to orchestrate or control the interactions between agents.
- ARION's runtime system uses mobile agents to facilitate the execution of remote tasks.
- A mobile agent's operation is supported by an agent platform (Grasshopper) that runs at every host that participates in the workflow execution to enable agent migration.

Agent Runtime technology

- Agents play the role of wrappers for task integration and interoperation.
- ARION is composed of a set of distributed nodes containing different data sets and programs (scientific collections).
- Workflows typically rely on distributed, autonomous tasks, and are controlled by a centralized server (ARION main server).
- The agents installed on each node execute workflow tasks, mainly computations, and monitor its services.

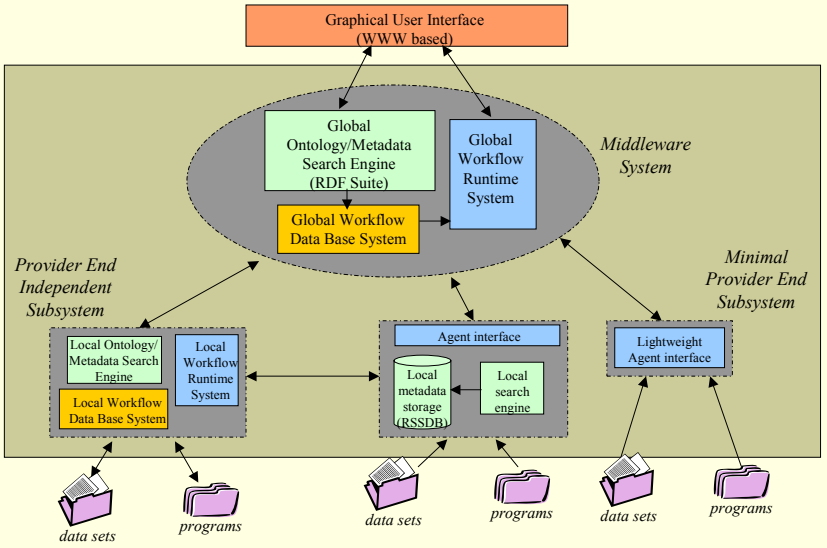
A Service Infrastructure for E-Science: the case of the ARION System
http://arion-dl.org

Lightweight system integration architecture

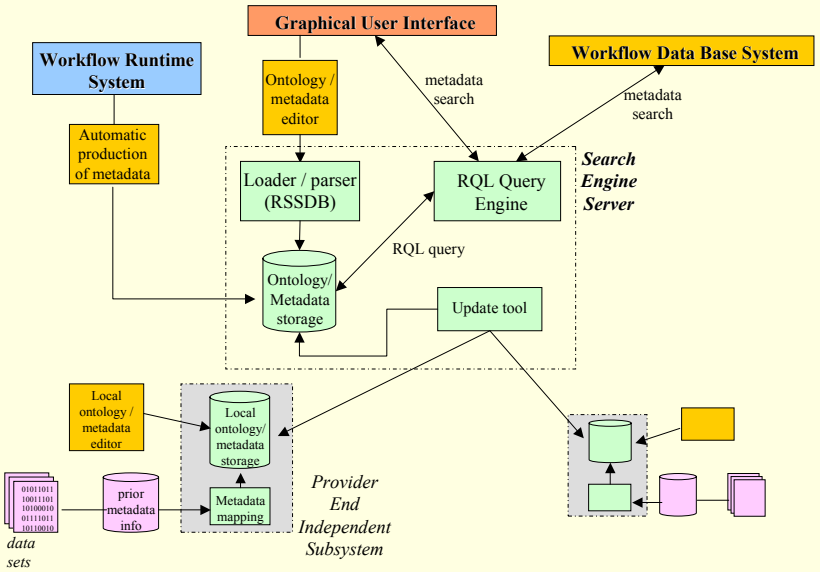
Consists of three independent parts:

- The *Search Engine,* which allows users to pose queries to the knowledge provided by ARION.
- The *Workflow Data Base System,* which contains the workflow specifications and handles the preparation of execution specifications to be sent to the workflow runtime system.
- The *Workflow Runtime System,* which is responsible for the execution of workflows and the management of the information produced during this execution from the distributed nodes.

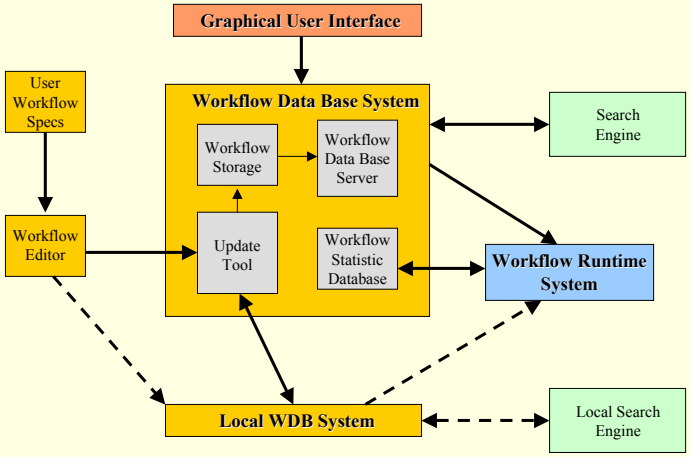
ARION System Architecture



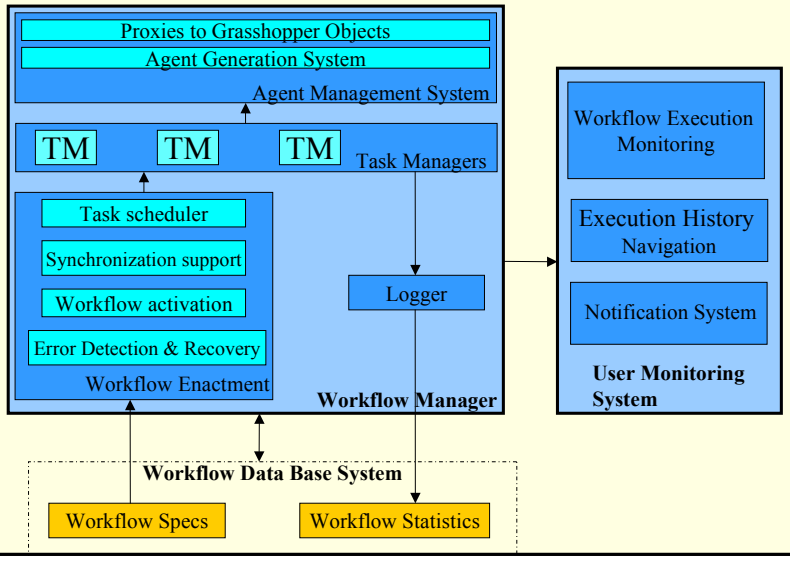
ARION Search Engine Architecture



Workflow Data Base System Architecture



Workflow Runtime System Architecture



Features of the architecture

- Scalable provider subsystems with tailored deployment
- Intuitive resource discovery
- On-line data set production
- User authentication and security
- Straightforward resource export procedure
- Lightweight system with minimal administration
- Metadata and data association techniques
- User interactive web access

A Service Infrastructure for E-Science: the case of the ARION System
http://arion-dl.org

Agent Runtime technology

- Supports
  - backtracking for parameter calibration
  - data shipping between processing resources
  - data/local resource security
  - monitoring of the computations

An example of competency questions

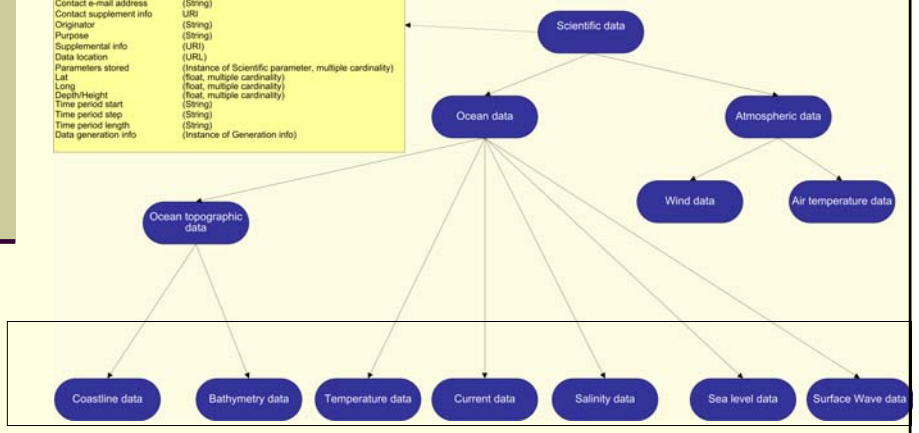
1. What kind of data do we have?
2. What parameters do our databases contain?
3. Which methodology has been followed to obtain the data? (e.g., measurements, simulations)
4. Which sensor has been used for the measured data?
5. What kind of models (physical or mathematical) has been used for the simulated data?

A Service Infrastructure for E-Science: the case of the ARION System
http://arion-dl.org

1. What kind of data do we have?

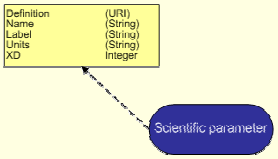
Publication date	(Date)
Form	URI
Format	(String)
Description of the format	URI
Delivery date	(Date)
Abstract	URI
Description	URI
Contact person	(String)
Contact address	(String)
Contact e-mail address	(String)
Contact supplement info	URI
Originator	(String)
Purpose	(String)
Supplemental info	(URI)
Data location	(URL)
Parameters stored	(Instance of Scientific parameter, multiple cardinality)
Lat	(float, multiple cardinality)
Long	(float, multiple cardinality)
Depth/Height	(float, multiple cardinality)
Time period start	(String)
Time period step	(String)
Time period length	(String)
Data generation info	(Instance of Generation info)

RQL query: ***leafclass***



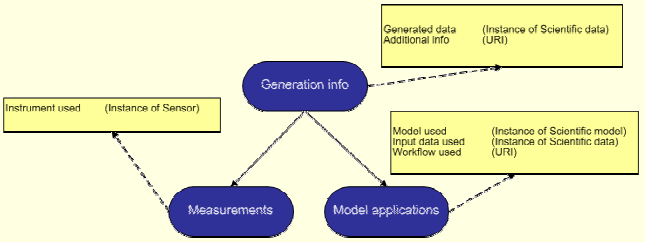
2. What parameters do our databases contain?

RQL query: ***select Y from {X;Scientific_Parameter}.NAME {Y}***



3. Which methodology has been followed to obtain the data?
(e.g. measurements, simulations)

RQL query: ***select Y from {X}Data_Generation_Info{Y} where X="DataSet1"***



A Service Infrastructure for E-Science: the case of the ARION System
http://arion-dl.org

Conclusions

- A scalable lightweight architecture of the ARION active repository has been presented.
- It provides the infrastructure for a scientific knowledge service for different domains of science.
- The repository provides ontology-based search service and workflow-based computational service.
- The user is capable of pose advanced queries and interacting with the computational service to customize his/her computations.
- ARION can be regarded as a tool for building advanced services, policy support, decision support, using information brokering techniques.

```
<?xml version="1.0" encoding="UTF-8"?>
<workflow ID="workflow1" name="Synthetic time series using Climar methodology">
    <sequence>
        <condition condition="User_data=Y">
            <true>
                <while_do condition="task1.result=N">
                    <task ID="task1">
                        <input>
                            <data-set ID="data-set1" name="User input time
series" address="" type="normal" domain=""/>
                            <parameter name="Lat" type="double"/>
                            <parameter name="Long" type="double"/>
                        </input>
                        <program name="Validation of user input" ID="program1"
address="" heavy="N" mobile="N" simulation="N" visualization="N" statistical="N"
validation="Y" database_retrieval="N" output_filter="N" transformation="N" domain=""/>
                        <output>
                            <data-set ID="data-set2" name="Source data"
address="" type="produced-storable" domain=""/>
                        </output>
                    </task>
                </while_do>
            </true>
```



```
            <false>
                <task ID="task2">
                    <input>
                        <data-set ID="data-set3" name="OCN Buoy
Database" address="" type="normal" domain=""/>
                        <data-set ID="data-set4" name="OCN UK Met. Office
Database" address="" type="normal" domain=""/>
                        <data-set ID="data-set5" name="NTUA WAM
Calibrated Database" address="" type="normal" domain=""/>
                        <parameter name="Lat" type="double"/>
                        <parameter name="Long" type="double"/>
                    </input>
                    <program name="Input source retrieval code"
ID="program2" address="" heavy="Y" mobile="Y" simulation="N"
visualization="N" statistical="N" validation="N" database_retrieval="Y"
output_filter="N" transformation="N" domain=""/>
                    <output>
                        <data-set ID="data-set6" name="source data"
address="" type="produced-storable" domain="" ref="data-set2"/>
                    </output>
                </task>
            </false>
        </condition>
```

```
<parallel_sync>
        <task ID="task3">
                <input>
                        <data-set ID="data-set7" name="input source data" address="" type="produced-
storable" domain="" ref="data-set2"/>
                </input>
                        <program name="Logarithmic Transformation Code" ID="program3" address=""
heavy="Y" mobile="Y" simulation="N" visualization="N" statistical="N" validation="N"
database_retrieval="N" output_filter="N" transformation="Y" domain=""/>
                <output>
                        <data-set ID="data-set8" name="Transformed time series" address=""
type="produced-storable" domain=""/>
                </output>
        </task>
        <sequence>
                <task ID="task4">
                        <input>
                                <data-set ID="data-set10" name="OCN Satellite Database"
address="" type="normal" domain=""/>
                                <data-set ID="data-set11" name="Input source data" address=""
type="produced-storable" domain="" ref="data-set2"/>
                        </input>
                                <program name="OCEANOR's Retreival Code" ID="program4"
address="" heavy="N" mobile="Y" simulation="N" visualization="N" statistical="N" validation="N"
database_retrieval="Y" output_filter="N" transformation="N" domain=""/>
                        <output>
                                <data-set ID="data-set9" name="Satellite Data" address=""
type="produced-storable" domain=""/>
                        </output>
                 </task>
```

A Service Infrastructure for E-Science: the case of the ARION System
http://arion-dl.org

```
                <task ID="task5">
                        <input>
                                <data-set ID="data-set12"
name="Satellite data" address="" type="produced-storable" domain=""
ref="data-set9"/>
                        </input>
                        <program name="Logarithmic Transformation
Code" ID="program5" address="" heavy="N" mobile="N" simulation="Y"
visualization="N" statistical="N" validation="N" database_retrieval="N"
output_filter="N" transformation="Y" domain=""/>
                        <output>
                                <data-set ID="data-set13"
name="Transformed Satelite Data" address="" type="produced-
storable" domain=""/>
                        </output>
                </task>
    </sequence>
   </parallel_sync>
 </sequence>
</workflow>
```