ENVRI Common Operations of Environmental Research Infrastructures

ENVRI

Services for the Environmental Community

D3.4 ENVRI Reference Model

Document identifier:	D3.4 ENVRI Reference Model
Date:	**30/04/2013**
Activity:	**WP3**
Lead Partner:	**CU**
Document Status:	**FINAL**
Dissemination Level:	**PUBLIC**
Document Link:	<link to the website>

ABSTRACT

It has been recognised that all ENVRI research infrastructures, although are very diverse, have some common characteristics, enabling them potentially to achieve a level of interoperability through the use of common standards for various functions. The objective of ENVRI Reference Model is to develop common ontological framework and standards for the description and characterisation of computational and storage infrastructures in order to achieve seamless interoperability between the heterogeneous resources of different infrastructures.

The ENVRI Reference Model is a work-in-progress, hosted by the ENVRI project, intended for interested parties to directly comment on and contribute to.

SEVENTH FRAMEWORK PROGRAMME

1. COPYRIGHT NOTICE

2. DELIVERY SLIP

	Name	Partner/Activity	Date
From			
Reviewed by	**Moderator:** **Reviewers:**	Leonardo Candela (CNR) Ari Asmi (UNEL)	22/04/13
Approved by			

3. DOCUMENT LOG

Issue	Date	Comment	Author/Partner
1.0	31/03/13	The draft version of the ENVRI Reference Model which describes an abstract model with a set of concepts and terms to capture a minimal sets of requirements of ESFRI environmental research infrastructures.	Yin Chen (CU) Paul Martine (UEDIN) Herbert Schentz (EAA) Barbara Magagna(EAA) Zhiming Zhao(UvA) Alex Hardisty (CU) Alun Preece (CU) Malcolm Atkinson (UEDIN)
2.0	30/04/13	Internally reviewed version to be approved by project management and submitted to the Commission.	Yin Chen (CU) Paul Martine (UEDIN) Herbert Schentz (EAA) Barbara Magagna(EAA) Zhiming Zhao(UvA)
3.0			
4.0			

SEVENTH FRAMEWORK PROGRAMME

4. APPLICATION AREA

This document is a formal deliverable for the European Commission, applicable to all members of the ENVRI project, beneficiaries and Joint Research Unit members, as well as its collaborating projects.

5. DOCUMENT AMENDMENT PROCEDURE

Amendments, comments and suggestions should be sent to the authors.

6. TERMINOLOGY

A complete project glossary is provided at the following page: http://www.ENVRI.eu/glossary. The terminology of concepts and terms defined in this document is provided in Appendix A.

7. PROJECT SUMMARY

Frontier environmental research increasingly depends on a wide range of data and advanced capabilities to process and analyse them. The ENVRI project, “Common Operations of Environmental Research infrastructures” is a collaboration in the ESFRI Environment Cluster, with support from ICT experts, to develop common e-science components and services for their facilities. The results will speed up the construction of these infrastructures and will allow scientists to use the data and software from each facility to enable multi-disciplinary science.

The target is on developing common capabilities including software and services of the environmental e-infrastructure communities. While the ENVRI infrastructures are very diverse, they face common challenges including data capture from distributed sensors, metadata standardisation, management of high volume data, workflow execution and data visualisation. The common standards, deployable services and tools developed will be adopted by each infrastructure as it progresses through its construction phase.

Two use cases, led by the most mature infrastructures, will focus the development work on separate requirements and solutions for data pre-processing of primary data and post-processing toward publishing.

The project will be based on a common reference model created by capturing the semantic resources of each ESFRI-ENV infrastructure. This model and the development driven by the test-bed deployments result in ready-to-use systems which can be integrated into the environmental research infrastructures.

The project puts emphasis on synergy between advanced developments, not only among the infrastructure facilities, but also with ICT providers and related e-science initiatives. These links will facilitate system deployment and the training of future researchers, and ensure that the inter-disciplinary capabilities established here remain sustainable beyond the lifetime of the project.

SEVENTH FRAMEWORK PROGRAMME

8. EXECUTIVE SUMMARY

This document describes the ENVRI Reference Model (ENVRI-RM) which is an abstract model with a set of concepts and terms capturing a set of requirements of environmental research infrastructures. Built on top of the Open Distributed Processing (ODP) framework, the Reference Model defines functional elements, data flow and dependencies that are common in ENVRI research infrastructures. The Reference Model can be used as the foundation for building reference architectures, and concrete implementations can be derived.

The ENVRI Reference Model is a work-in-progress, hosted by the ENVRI project, intended for interested parties to directly comment on and contribute to.

9. HOW TO READ

The document is organised as follows:

Section 1 introduces the motivation and background knowledge of the ENVRI-RM.

Section 2 presents an overview of the ENVRI-RM, which consists of 5 common subsystems identified in a pre-study work. The concepts of these entities and their relationship is discussed.

Section 3 detailed describes the ENVRI Reference Model from ODP three Viewpoints, the Science, Information and Computational.

Section 4 concludes this work.

Appendixes are not part of the model, and provided for the convenience of the reader.

Appendix A is the glossary of the document which consists of all concepts and terms defined throughout the ENVRI-RM.

Appendix B presents the full list of the required functionalities which is the result of the investigation of the common requirements of ENVRI Research Infrastructures.

Appendix C provides detailed dynamic schemata specified in Information Viewpoint (Section 4.2.3).

The intended audience of this document is the ENVRI community as well as other organisations or individuals that are interested in understanding the top level technical architecture which underpins the construction of such an architecture. In particular, the intended primary audience of this documents includes:

- Research Infrastructure implementation teams:
    - Architects, designers, and integrators;

    - Engineers – to enable them to be able to drill down directly to find required knowledge.
- Research Infrastructure Operations Teams, and
- Third party solution or component providers

The documents is also intended for Research Infrastructure leaders and Service Centre staffs.

The document can be read by others who want to better understand the ENVRI ongoing work, to gain understanding necessary to make contributions to the standardisation processes of environmental research infrastructures.

For the primary audience of the ENVRI-RM shall read the whole document.

For the leaders of research infrastructures, service centre staffs may want to read the introduction and background knowledge in **section 1** and model overview in **section 2**.

For readers who have general interests of the ENVRI reference model may want to read the **section 1** introduction.

TABLE OF CONTENTS

TABLE OF FIGURES

1 INTRODUCTION

1.1 Purpose and Scope

It has been recognised that all ENVRI research infrastructures, although very diverse, have some common characteristics, enabling them potentially to achieve a greater level of interoperability through the use of common standards for various functions. The objective of the ENVRI Reference Model is to develop a common ontological framework and standards for the description and characterisation of computational and storage infrastructures in order to achieve seamless interoperability between the heterogeneous resources of different infrastructures.

The ENVRI Reference Model serves the following purposes [1]:

- to provide a way for structuring thinking which helps the community to reach a common vision;
- to provide a common language which can be used to communicate concepts concisely;
- to help discover existing solutions to common problems;
- to provide a framework into which different functional components of research infrastructures can be placed, in order to draw comparisons and identify missing functionality.

This document describes the ENVRI Reference Model which:

- captures computational characteristics of data and operations that are common in ENVRI Research Infrastructures;
- establishes a taxonomy of terms, concepts and definitions to be used by the ENVRI community.

The Reference Model provides an abstract conceptual model; it does not impose a specific architecture nor does it impose any specific design decisions on the design of an infrastructure.

The *initial* model focuses on the urgent and important issues prioritised for ENVRI research infrastructures including data preservation, discovery and access, and publication. It defines a minimal set of computational functionalities to support these requirements. The core set will be extended incrementally over the course of the ENVRI project. The initial model does not cover engineering mechanisms or the applicability of existing standards or technologies.

1.2 Rationale

Environmental issues will dominate the 21st century [2]. Research infrastructures which provide advanced capabilities for data sharing, processing and analysis enable excellent research and play an ever-increasing role in the environmental sciences. The ENVRI project gathers 6 EU ESFRI [1] environmental infrastructures (ICOS[2], EURO-Argo[3], EISCAT-3D[4], LifeWatch[5], EPOS[6], and EMSO[7])

[1] ESFRI, the European Strategy Forum on Research Infrastructures, is a strategic instrument to develop the scientific integration of Europe and to strengthen its international outreach.
[2] ICOS, http://www.icos-infrastructure.eu/, is a European distributed infrastructure dedicated to the monitoring of greenhouse gases (GHG) through its atmospheric, ecosystem and ocean networks.
[3] EURO-Argo, http://www.euro-argo.eu/, is the European contribution to Argo, which is a global ocean observing system.
[4] EISCAT-3D, http://www.eiscat3d.se/, is a European new-generation incoherent-scatter research radar for upper atmospheric science.
[5] LifeWatch, http://www.lifewatch.com/, is an e-science Infrastructure for biodiversity and ecosystem research.

SEVENTH FRAMEWORK PROGRAMME

in order to develop common data and software services. The results will accelerate the construction of these infrastructures and improve interoperability among them by encouraging the adoption of the reference model. The experiences gained from this endeavour will also benefit the building of other advanced research infrastructures.

The primary objective of ENVRI is to agree on a reference model for joint operations. This will enable greater understanding and cooperation between users since fundamentally the model will serve to provide a universal reference framework for discussing many common technical challenges facing all of the ESFRI-ENV infrastructures. By drawing analogies between the reference components of the model and the actual elements of the infrastructures (or their proposed designs) as they exist now, various gaps and points of overlap can be identified [3].

The ENVRI Reference Model is based on the design experiences of the state-of-the-art environmental research infrastructures, with a view of informing future implementation. It tackles multiple challenging issues encountering many existing initiatives, such as data streaming and storage management; data discovery and access to distributed data archives; linked computational, network and storage infrastructure; data curation, data integration, harmonisation and publication; data mining and visualisation, and scientific workflow management and execution. It uses Open Distributed Processing (ODP), a standard framework for distributed system specification, to describe the model.

To our best knowledge there is no existing reference model for environmental science research infrastructures. This work intends to make a first attempt, which can serve as a basis to inspire future research explorations.

There is an urgent need to create such a model, as we are at the beginning of a new era. The advances in automation, communication, sensing and computation enable experimental scientific processes to generate data and digital objects at unprecedentedly great speeds and volumes. Many infrastructures are starting to be built to exploit the growing wealth of scientific data and enable multi-disciplinary knowledge sharing. In the case of ENVRI, most investigated RIs are in their planning / construction phase. The high cost attached to the construction of environmental infrastructures require cooperation on the sharing of experiences and technologies, solving crucial common e-science issues and challenges together. Only by adopting a good reference model can the community secure interoperability between infrastructures, enable reuse, share resources and experiences, and avoid unnecessary duplication of effort.

The contribution of this work is threefold:

- The model captures the computational requirements and the state-of-the-art design experiences of a collection of representative research infrastructures for environmental sciences. It is the first reference model of this kind which can be used as a basis to inspire future research.
- It provides a common language for communication to unify understanding. It serves as a community standard to secure interoperability.
- It can be used as a base to drive design and implementation. Common services can be provided which can be widely applicable to various environmental research infrastructures and beyond.

[6] EPOS, http://www.epos-eu.org/, is a European Research Infrastructure on earthquakes, volcanoes, surface dynamics and tectonics.
[7] EMSO, http://www.emso-eu.org/management/, is a European network of seafloor observatories for the long-term monitoring of environmental processes related to ecosystems, climate change and geo-hazards.

1.3 Basis

The ENVRI Reference Model is built on top of the Open Distributed Processing (ODP) framework [4-7]. ODP is an international standard for architecting open, distributed processing systems. It provides an overall conceptual framework for building distributed systems in an incremental manner.

The reasons for adopting the ODP framework in the ENVRI project come from three aspects:

- It enables large collaborative design activities;
- It provides a framework for specifying and building large or complex system which consists of a set of guiding concepts and terminology. This provides a way of thinking about architectural issues in terms of fundamental patterns or organising principles; and
- Being an international standard, ODP offers authority and stability.

ODP adopts the ***object modelling*** approach to system specification. ISO/IEC 10746-2 [5] includes the formal definitions of the concepts and terminology adopted from object models, which provide the foundation for expressing the architecture of ODP systems. The modelling concepts fall into three categories [4, 5]:

- Basic modelling concepts for a general object-based model;
- Specification concepts to allow designers to describe and reason about ODP system specifications;
- Structuring concepts, including organisation, the properties of systems and objects, management, that correspond to notions and structures that are generally applicable in the design and description of distributed systems.

ODP is best known for its use of viewpoints. A *viewpoint* (on a system) is an abstraction that yields a specification of the whole system related to a particular set of concerns. The ODP reference model defines five specific viewpoints as follows [4, 6]:

- The ***Enterprise Viewpoint***, which concerns the organisational situation in which business (research activity in the current case) is to take place. In order to better communicate with the ENVRI community, in this document, we rename it as **Science Viewpoint**;
- The ***Information Viewpoint***, which concerns modelling of the shared information manipulated within the system of interest;
- The ***Computational Viewpoint***, which concerns the design of the analytical, modelling and simulation processes and applications provided by the system;
- The ***Engineering Viewpoint***, which tackles the problems of diversity in infrastructure provision; it gives the prescriptions for supporting the necessary abstract computational interactions in a range of different concrete situations;
- The ***Technology Viewpoint***, which concerns real-world constraints (such as restrictions on the facilities and technologies available to implement the system) applied to the existing computing platforms on which the computational processes must execute.

This version of the ENVRI Reference Model covers 3 ODP viewpoints: the science, information, and computational viewpoints.

SEVENTH FRAMEWORK PROGRAMME

1.4 Approaches

The approach leading to the creation of the ENVRI Reference Model is based on the analysis of the requirements of a collection of representative environmental research infrastructures, which are reported in two ENVRI deliverable:

- D3.1:Assesment of the State of the Art
- D3.3: Analysis of Common Requirements for ENVRI Research Infrastructures

The ODP standard is used as the modelling and specification framework, which enables the designers from different organisations to work independently and collaboratively.

The development starts from a core model and will be incrementally extended based on the community common requirements and interests.

The reference model will be evaluated by examining the feasibilities in implementations, and the refinement of the model will be based on community feedback.

1.5 Conformance

A conforming environmental research infrastructure should support the common subsystems described in section 3 and the functional and information model described in section 4.

The ENVRI Reference Model does not define or require any particular method of implementation of these concepts. It is assumed that implementers will use this reference model as a guide while developing a specific implementation to provide identified services and content. A conforming environmental research infrastructure may provide additional services to users beyond those minimally required computations defined in this document.

Any descriptive (or prescriptive) documents that claim to be conformant to the ENVRI Reference Model should use the terms and concepts defined herein in a similar way.

1.6 Related Work

1.6.1 Related concepts

A **reference model** is an abstract framework for understanding significant relationships among the entities of some environment. It consists of a minimal set of unifying concepts, axioms and relationships within a particular problem domain. [8]

A reference model is not a reference architecture. A **reference architecture** is an architectural design pattern indicating an abstract solution that implements the concepts and relationships identified in the reference model [8]. Different from a reference architecture, a reference model is independent from specific standards, technologies, implementations or other concrete details. A reference model can drive the development of a reference architecture or more than one of them [9].

It could be argued that a reference model is, at its core, an **ontology**. Conventional reference models e.g., OSI[10], RM-ODP [4], OAIS[11] , are built upon modelling disciplines. Many recent works, such as the DL.org Digital Library Reference Model [9], are more ontology-like.

Both models and ontologies are technologies for information representation, but have been developed separately in different domains. Modelling approaches have risen to prominence in the software engineering domain over the last ten to fifteen years [12]. Traditionally, software engineers have taken very pragmatic approaches to data representation, encoding only the information needed to solve the problem in hand, usually in the form of language data structures or database tables. Modelling approaches are meant to increase the productivity by maximising compatibility between systems (by reuse of standardised models), simplifying the process of design (by models of recurring design patterns in the application domain), and promoting communication between individuals and teams working on the system (by a standardisation of the terminology and the best practices used in the application domain) [13]. On the other hand, ontologies have been developed by the Artificial Intelligence community since the 1980s. An ontology is a structuring framework for organising information. It renders shared vocabulary and taxonomies which models a domain with the definition of objects and concepts and their properties and relations. These ideas have been heavily drawn upon in the notion of the Semantic Web. [13]

Traditional views tend to distinguish the two technologies. The main points of argument include but are not limited to:

1. Models usually focus on realisation issues (e.g., the Object-Oriented Modelling approach), while ontologies usually focus on capturing abstract domain concepts and their relationship [14].
2. Ontologies are normally used for run-time knowledge exploitation (e.g., for knowledge discovery in a knowledgebase), but models normally do not [15].
3. Ontologies can support reasoning while models cannot (or do not) [13].
4. Finally, models are often based on the Closed World Assumption while ontologies are based on the Open World Assumption [13].

However, these separations between the two technologies are rapidly disappearing in recent developments. Study [13] shows that ‘all ontologies are models’, and ‘almost all models used in modern software engineering qualify as ontologies.’ As evidenced by the growing number of research workshops dealing with the overlap of the two disciplines (e.g., SEKE [16], VORTE [17], MDSW [18], SWESE [19], ONTOSE [20], WoMM [21]), there has been considerable interests in the integration of software engineering and artificial intelligence technologies in both research and practical software engineering projects.[13]

We tend to take this point of view and regard the ENVRI Reference Model as both a model and an ontology. The important consequence is that we can explore further in both directions, e.g., the reference model can be expressed using a modelling language, such as UML (UML4ODP). It can then be built into a tool chain, e.g., to plugin to an integrated development environment such as Eclipse, which makes it possible to reuse many existing UML code and software. On the other hand, the reference model can also be expressed using an ontology language such as RDF or OWL which can then be used in a knowledge base. In this document we explore principally from modelling aspects. In another ENVRI task, T3.4, the ontological aspect of the reference model will be exploited.

Finally, a reference model is a **standard**. Created by ISO in 1970, OSI is probably among the earliest reference models, which defines the well-known 7-layered network communication. As one of the ISO standard types, the reference model normally describes the overall requirements for standardisation and the fundamental principles that apply in implementation. It often serves as a framework for more

specific standards [22]. This type of standard has been rapidly adopted, and many reference models exist today, which can be grouped into 3 categories, based on the type of agreement and the number of people, organisations or countries who were involved in making the agreement:

- **Committee reference model** – a widely-based group of experts nominated by organizations who have an interest in the content and application of the standard build the standard.
- **Consensus reference model** – the principle that the content of the standard is decided by general agreement of as many as possible of the committee members, rather than by majority voting. The ENVRI Reference Model falls into this group.
- **Consultation reference model** – making a draft available for scrutiny and comment to anyone who might be interested in it.

Some examples from each of the categories are discussed below, with emphasis on approaches and technologies.

1.6.2 Related reference models

1.6.2.1 Committee Reference Models

In this category, we look at those defined by international organisations , such as the Advancing Open Standards for the Information Society (OASIS), the Consultative Committee for Space Data Systems (CCSDS), and the Open Geospatial Consortium (OGC).

The Open Archival Information System (OAIS) Reference Model [11] is an international standard created by CCSDS and ISO which provides a framework, including terminology and concepts for archival concept needed for Long-Term digital information preservation and access.

The OASIS Reference Model for Service Oriented Architecture (SOA-RM) [8] defines the essence of service oriented architecture emerging with a vocabulary and a common understanding of SOA. It provides a normative reference that remains relevant to SOA as an abstract model, irrespective of the various and inevitable technology evolutions that will influence SOA deployment.

The OGC Reference Model (ORM) [23], describes the OGC Standards Baseline, and the current state of the work of the OGC. It provides an overview of the results of extensive development by OGC Member Organisations and individuals. Based on RM-ODP's 5 viewpoints, ORM captures business requirements and processes, geospatial information and services, reusable patterns for deployment, and provides a guide for implementations.

The Reference Model for the ORCHESTRA Architecture (RM-OA) [24] is another OGC standard. The goal of the integrated project ORCHESTRA (Open Architecture and Spatial Data Infrastructure for Risk Management) is the design and implementation of an open, service-oriented software architecture to overcome the interoperability problems in the domain of multi-risk management. The development approach of RM-OA is standard-based which is built on the integration of various international standards. Also using RM-ODP standard as the specification framework, RM-OA describes a platform neutral (abstract) model consisting of the informational and functional aspects of service networks combining architectural and service specification defined by ISO, OGC, W3C, and OASIS. [24]

1.6.2.2 Consensus Reference Models

In this category, we discuss those created by non-formal standard organisations.

The LifeWatch Reference Model [25], developed by the EU LifeWatch consortium, is a specialisation of the RM-OA standard which provides the guidelines for the specification and implementation of a biodiversity research infrastructure. Inherited from RM-OA, the reference model uses the ODP standard as the specification framework.

The Digital Library Reference Model [9] developed by DL.org consortium introduces the main notations characterising the whole digital library domain, in particular, it defines 3 different types of systems: (1) Digital Library, (2) Digital Library System, and (3) Digital Library Management System; 7 core concepts characterising the digital library universe: (1) Organisation, (2) Content, (3) Functionality, (4) User, (5) Policy, (6) Quality, and (7) Architecture; and 3 categories of actors: (1) DL End-Users (including, Content Creators, Content Consumers, and Digital Librarians), (2) DL Managers (Including, DL Designer, and DL System Administrators) , and (3) DL Software Developers.

The Workflow Reference Model [26] provides a common framework for workflow management systems, identifying their characteristics, terminology and components. The development of the model is based on the analysis of various workflow products in the market. The workflow Reference Model firstly introduces a top level architecture and various interfaces it has which may be used to support interoperability between different system components and integration with other major IT infrastructure components. This maps to the ODP Computational Viewpoint. In the second part, it provides an overview of the workflow application program interface, comments on the necessary protocol support for open interworking and discusses the principles of conformance to the specifications. This maps to the ODP Technology Viewpoint.

The Agent System Reference Model [27] provides a technical recommendation for developing agent systems, which captures the features, functions and data elements in the set of existing agent frameworks. Different from conventional methods, a reverse engineering method has been used to develop the reference model, which starts by identifying or creating an implementation-specific design of the abstracted system; secondly, identifying software modules and grouping them into the concepts and components; and finally, capturing the essence of the abstracted system via concepts and components.

1.6.2.3 Consultation Reference Models

The Data State Reference Model [28] provides an operator interaction framework for visualisation systems. It breaks the visualisation pipeline (from data to view) into 4 data stages (Value, Analytical Abstraction, Visualisation Abstraction, and View), and 3 types of transforming operations (Data Transformation, Visualisation Transformation and Visual Mapping Transformation). Using the data state model, the study [29] analyses 10 existing visualisation techniques including, 1) scientific visualisations, 2) GIS, 3) 2D, 4) multi-dimensional plots, 5) trees, 6) network, 7) web visualisation, 8) text, 9) information landscapes and spaces, and 10) visualisation spread sheets. The analysis results in a taxonomy of existing information visualisation techniques which help to improve the understanding of the design space of visualisation techniques.

The Munich Reference Model [30] is created for adaptive hypermedia applications which is a set of nodes and links that allows one to navigate through the hypermedia structure and that dynamically

"adapts" (personalise) various visible aspects of the system to individual user's needs. The Munich Reference Model uses an object-oriented formalisation and a graphical representation. It is built on top of the Dexter Model layered structure, and extends the functionality of each layer to include the user modelling and adaptation aspects. The model is visually represented using in UML notation and is formally specified in Object Constraint Language (which is part of the UML).

While these works use a similar approach to the development of the reference model as the ENVRI-RM, which is based on the analysis of existing systems and abstracts to obtain the 'essence' of those systems, a major difference is that these works have not normally met with significant feedback or been formally approved by an existing community, with the consequence that they express less authority as a standard.

1.6.3 Other Related standards

Data Distribution Service for Real-Time Systems (DDS) [31], an Object Management Group (OMG) standard, is created to enable scalable, real-time, dependable, high performance, interoperable data exchanges between publishers and subscribers. DDS defines a high-level conceptual model as well as a platform-specific model. UML notations are used for specification. While DDS and the ENVRI share many similar views in design and modelling, DDS focuses on only one specific issue, i.e., to model the communication patterns for real-time applications; while ENVRI aims to capture a overall picture of requirements for environmental research infrastructures.

Published by the web standards consortium OASIS in 2010, the Content Management Interoperability Services (CMIS) [32] is an open standard that allows different content management systems to inter-operate over the Internet. Specially, CMIS defines an abstraction layer for controlling diverse document management systems and repositories using web protocols. It defines a domain model plus web services and Restful AtomPub bindings that can be used by applications to work with one or more Content Management repositories/systems. However as many other OASIS standards, CMIS is not a conceptual model and is highly technology dependent [32].

In the next, we introduce the ENVRI Reference Model.

2 MODEL OVERVIEW

2.1 Subsystems of Environmental Infrastructures

In a pre-study, we have investigated a collection of representative environmental research infrastructures. By examining their computational characteristics, we have identified 5 common subsystems: *Data Acquisition*, *Data Curation*, *Data Access*, *Data Processing* and *Community Support*. The fundamental reason of the division of the 5 subsystems is based on the observation that all applications, services and software tools are designed and implemented around 5 major physical resources: the sensor network, the storage, the (internet) communication network, application servers and client devices.

The ENVRI Reference Model is partitioned into the five subsystems. The partitioning of the reference model into subsystems is based broadly on a notion of data life-cycle evident in all existing research infrastructures investigated plus a generic one dedicated to community management.

This lifecycle begins with the acquisition of raw data from a network of integrated data collecting instruments (seismographs, weather stations, robotic buoys, human observations, etc.) which is then pre-processed and curated within a number of data stores belonging to an infrastructure or one of its delegate infrastructures. This data is then made accessible to authorised requests by parties outwith the infrastructure, as well as to services within the infrastructure. This results in a nature partitioning of data acquisition, curation and access. In addition, data can be extracted from parts of the infrastructure and made subject to data processing, the results of which can then be resituated within the infrastructure. Finally, the community support subsystem provides tools and services required to handle data outside of the core infrastructure and reintegrate it when necessary.

Each subsystem should provide a set of capabilities via interfaces invoked by the other subsystems. In ODP, an interface is simply an abstraction of the behaviour of an object that consists of a subset of the interactions expected of that object together with the constraints imposed on their occurrence.

2.1.1 Data Acquisition

The ***data acquisition subsystem*** of a research infrastructure collects raw data from registered sources to be stored and made accessible within the infrastructure.

The data acquisition subsystem collects raw data from sensor arrays and other instruments, as well as from human observers, and brings those data into the system. Within the ENVRI-RM, the acquisition subsystem is considered to begin upon point of data entry into the modelled system, the general maintenance and deployment of sensor stations and human observers being outside the scope of ENVRI. Acquisition is typically distributed across a network of observatories and stations. Data acquired is generally assumed to be non-reproducible, being associated with a specific (possibly continuous) event in time and place; as such, the assignment of provenance (particularly data source and timestamp) is essential. Real-time data streams may be temporarily stored, sampled, filtered and processed (e.g., based on applied quality control criteria) before being passed on for curation. Control software is often deployed to manage and schedule the execution and monitoring of data flows. Data collected by the acquisition subsystem must ultimately be transferred to the data curation subsystem for preservation, usually within a specific time period.

2.1.2 Data Curation

The ***data curation subsystem*** of a research infrastructure stores, manages and ensures access to all persistent data-sets produced within the infrastructure.

The data curation subsystem facilitates quality control and preservation of scientific data. The subsystem is typically implemented across one or more dedicated data centres. Data handled by the subsystem include raw data products, metadata and processed data; where possible, processed data should be reproducible by executing the same process on the same source data-sets. Operations such as data quality verification, identification, annotation, cataloguing, replication and archival are often provided. Access to curated data from outside the infrastructure is brokered through the data access subsystem. There is usually an emphasis on non-functional requirements for data curation satisfying availability, reliability, utility, throughput, responsiveness, security and scalability criteria.

2.1.3 Data Access

The ***data access subsystem*** of a research infrastructure enables discovery and retrieval of scientific data subject to authorisation.

The data access subsystem enables discovery and retrieval of data housed in data resources managed by the data curation subsystem. Data access subsystems often provide gateways for presenting or delivering data products. Query and search tools may be provided which allow users or upstream services to discover data based on metadata or semantic linkages. Data handled by the access subsystem need not be homogeneous. When supporting heterogeneous data, different types of data (often pulled from a variety of distributed data resources) may be converted into uniform representations with uniform semantics resolved by a data discovery service. The subsystem may provide services for harvesting, compressing and packaging (meta)data as well as encoding services for secure data transfer. Data access is controlled using authentication and authorisation policies. Despite the name, the access subsystem may also provide services for importing data into the infrastructure.

2.1.4 Data Processing

The ***data processing subsystem*** of a research infrastructure provides a toolbox of services for performing a variety of data processing tasks.

The data processing subsystem is able to aggregate data from various sources and conduct a range of experiments and analyses upon that data. Data handled by the subsystem are typically derived and recombined via the data access subsystem. The data processing subsystem is expected to offer operations for statistical analysis and data mining as well as facilities for conducting scientific experiments, modelling and simulation, and scientific visualisation. Performance requirements for processing scientific data tend to be concerned with scalability which may be addressable at the level of engineering (e.g., by making use of Grid or Cloud services).

2.1.5 Community Support

The ***community support subsystem*** of a research infrastructure exists to support users of an infrastructure in their interactions with that infrastructure.

The community support subsystem manages, controls and tracks users' activities and supports users to conduct their roles in their communities. Data 'handled' within the subsystem are typically user-generated data and communications. The community support subsystem may support interactive visualisation, standardised authentication, authorisation and accounting protocols, and the use of virtual organisations. The subsystem is considered to encircle the other four subsystems, describing the interface between the research infrastructure and the wider world in which it exists.

2.2 Subsystem Relationships

As shown in Figure 3.1, amongst the five subsystems can be identified seven major points-of-reference wherein interfaces between subsystems can be implemented.

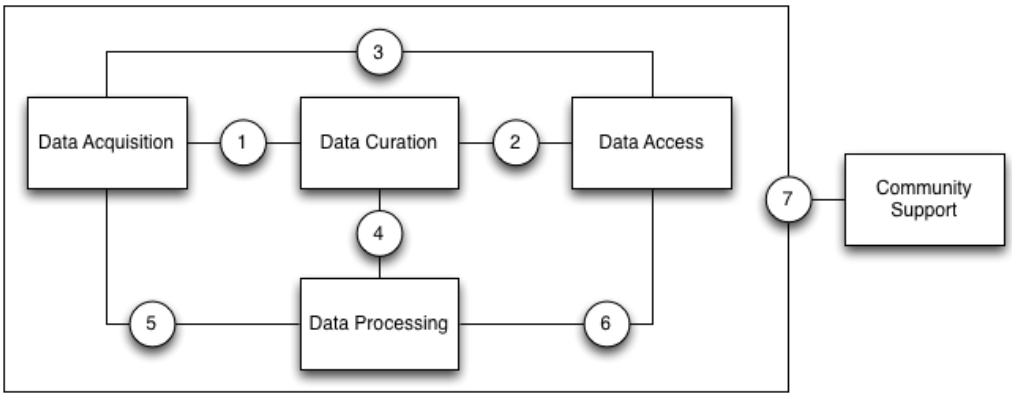


Figure 3.1: Illustration of the major points-of-reference between different subsystems of the ENVRI-RM.

These points-of-reference are as follows:

1. **Acquisition/Curation** by which the collection of raw data is managed.
2. **Curation/Access** by which the retrieval of curated data products is arranged.
3. **Acquisition/Access** by which the status of the data acquisition network can be monitored externally.
4. **Curation/Processing** by which analyses of curated data is coordinated.
5. **Acquisition/Processing** by which acquisition events are listened for and responded to.
6. **Processing/Access** by which data processes are scheduled and reported.
7. **Community/All** by which the outside world interacts with the infrastructure in many different roles.

Depending on the distribution of resources in an implemented infrastructure, some of these reference points may not be present in the infrastructure. They take particular importance however when considering scenarios where a research infrastructure delegates subsystems to other client infrastructures. For example, EPOS and LifeWatch both delegate data acquisition and some data curation activities to client national or domain-specific infrastructures, but provide data processing services over the data held by those client infrastructures. Thus reference points 4 and 5 become of great importance to the construction of those projects.

2.3 Common Functions within Common Subsystems

Analysis of the common requirements of the six ESFRI environmental infrastructures affiliated with the ENVRI project has resulted in the identification of a number of common functionalities. These functionalities can be partitioned amongst the five subsystems of the ENVRI-RM and presented as interfaces of each subsystem. They encompass a range of concerns, from the fundamental (*e.g.* data collection and storage, data discovery and access and data security) to more specific challenges (*e.g.* data versioning, instrument monitoring and interactive visualisation).

Figure 3.2: Radial depiction of ENVRI-RM requirements with the minimal model highlighted.

In order to better manage the range of requirements, and in order to ensure rapid publication of incremental refinements to the ENVRI-RM, as highlighted in Figure 3.2, a *minimal model* has been identified which describes the fundamental functionality necessary to describe a functional environmental research infrastructure. By initially focusing on this minimal model, it then becomes practical to produce a partial specification of the ENVRI-RM which nonetheless reflects the final shape of the ENVRI-RM without the need for significant refactoring. Further development of the ENVRI-RM will focus on designated priority areas based on feedback from the contributing ESFRI representatives.

The definitions of the minimal set of functions are given as follows. The definition of the full list of common functions are provided in Appendix A.

(A) Data Acquisition Subsystem

Process Control: A functionality that receives input status, applies a set of logic statements or control algorithms, and generates a set of analogue / digital outputs to change the logic states of devices.

Data Collection: A functionality that obtains digital values from a sensor instrument, associating consistent timestamps and necessary metadata.

Data Transmission: A functionality that transfers data over a communication channel using specified network protocols.

(B) Data Curation Subsystem

Data Quality Checking: A functionality that detects and corrects (or remove) corrupt, inconsistent or inaccurate records from datasets.

Data Identification: A functionality that assigns (global) unique identifiers to data contents.

Data Cataloguing: A functionality that associates a data object with one or more metadata objects which contain data descriptions.

Data Product Generation: A functionality that processes data against requirement specifications and standardised formats and descriptions.

Data Storage & Preservation: A functionality that deposits (over the long-term) data and metadata or other supplementary data and methods according to specified policies, and then to make them accessible on request.

(C) Data Access Subsystem

Access Control: A functionality that approves or disapproves of access requests based on specified access policies.

Metadata Harvesting: A functionality that (regularly) collects metadata in agreed formats from different sources.

Resource Registration: A functionality that creates an entry in a resource registry and inserts a resource object or a reference to a resource object with specified representation and semantics.

Data Publication: A functionality that provides clean, well-annotated, anonymity-preserving datasets in a suitable format, and by following specified data-publication and sharing policies to make the datasets publically accessible or to those who agree to certain conditions of use, and to individuals who meet certain professional criteria.

Data Citation: A functionality that assigns an accurate, consistent and standardised reference to a data object, which can be cited in scientific publications.

Data Discovery and Access: A functionality that retrieves requested data from a data resource by using suitable search technology.

(D). Data Processing Subsystem

Data Assimilation: A functionality that combines observational data with output from a numerical model to produce an optimal estimate of the evolving state of the system.

Data Analysis: A functionality that inspects, cleans, transforms data, and to provide data models with the goal of highlighting useful information, suggesting conclusions, and supporting decision making.

Data Mining: A functionality that supports the discovery of patterns in large datasets.

Data Extraction: A functionality that retrieves data out of (unstructured) data sources, including web pages, emails, documents, PDFs, scanned text, mainframe reports, and spool files.

Scientific Modelling and Simulation: A functionality that supports of the generation of abstract, conceptual, graphical or mathematical models, and to run an instance of the model.

(Scientific) Workflow Enactment: A specialisation of Workflow Enactment, which support of composition and execution a series of computational or data manipulation steps, or a workflow, in a scientific application. Important processes should be recorded for provenance purposes.

Data Processing Control: A functionality that initiates the calculation and manage the outputs to be returned to the client.

(E) Community Support Subsystem

Authentication: A functionality that verifies the credentials of a user.

Authorisation: A functionality that specifies access rights to resources.

3 ENVRI REFERENCE MODEL

The ENVRI Reference Model is structured according to the Open Distributed Processing (ODP) standard. As such, the Reference Model is defined from five different perspectives. In the context of ENVRI, which uses ODP to define an 'archetypical' environmental research infrastructure rather than a specific (implemented) infrastructure, three viewpoints take particular priority – the *Science*, *Information* and *Computational* viewpoints.

The remaining two viewpoints (*Engineering* and *Technology*) are more relevant to specific instances of research infrastructure. Nevertheless, the ENVRI Reference Model will address these viewpoints to some extent in future revisions.

3.1 Science Viewpoint

The Science Viewpoint of the ENVRI-RM intends to capture the requirements for an environmental research infrastructure from the perspective of the people who perform their tasks and achieve their goals as mediated by the infrastructure. Modelling in this viewpoint uses a reverse engineering method, which derives the principles and properties of model objects through the analysis of the structure and functionality of the real-world systems.

In a pre-study, we have observed 5 subsystems commonly exist in environmental science research infrastructures: *Data Acquisition*, *Data Curation*, *Data Access*, *Data Processing* and *Community Support*. Correspondingly, human activities which interact with the 5 subsystems in order to collaboratively conduct scientific research, from data collection to the deliverance of scientific results, can also be grouped in the same way. Such groups are the so-called *communities* in ODP. In this viewpoint, we examine what those communities are, what kind of roles they have, and what main behaviours they act out.

3.1.1 Common Communities

A **community** is a collaboration which consists of a set of *roles* agreeing their objective to achieve a stated business purpose.

In the ENVRI-RM, we distinguish 5 activities, seen as communities in accordance to the 5 common sub-systems. As shown in Figure 4.1, the 5 communities are, *data acquisition*, *data curation*, *data publication*, *data service provision*, and *data usage* community. The definition of the communities are based on their objectives.

- **Data Acquisition Community**, who collects raw data and brings (streams of) measures into a system;
- **Data Curation Community**, who curates the scientific data, maintains and archives them, and produces various data products with metadata;
- **Data Publication Community**, who assists data publication, discovery and access;
- **Data Service Provision Community**, who provides various services, applications and software/tools to link and recombine data and information in order to derive knowledge;
- **Data Usage Community**, who make use of data and service products, and transfer knowledge into understanding.

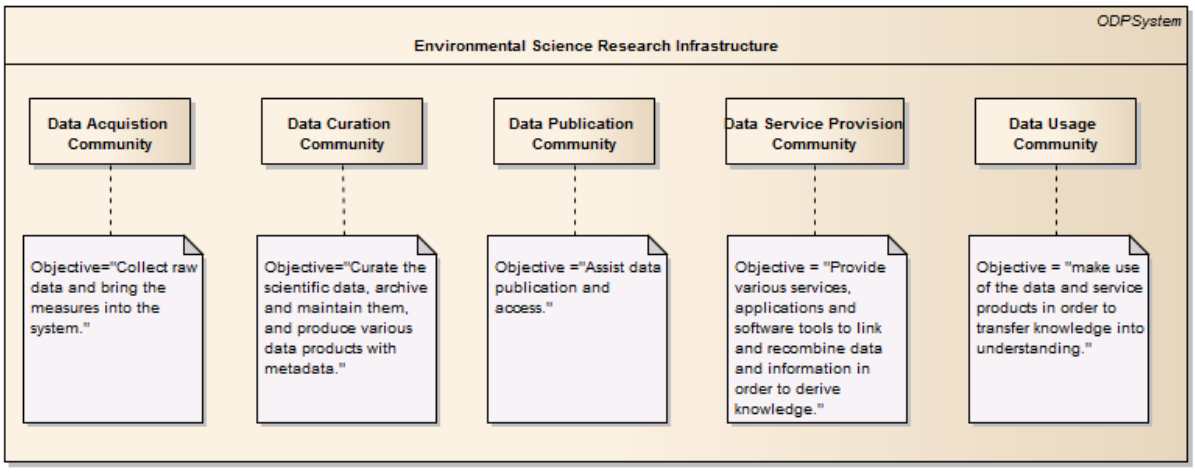


Figure 4.1: Common Communities

3.1.2 Common Community Roles

A **role** in a community is a prescribing behaviour that can be performed any number of times concurrently or successively. A role can be either *active* (typically associated with a human actor) or *passive* (typically associated with a non-human actor).

In the following, we identify *active roles* in relation to people associated with a research infrastructure:

- those who use the research infrastructure to do science;
- those who work on resources to build, maintain and operate the research infrastructure; and
- those who govern, manage and administer the research infrastructure

Note, an individual may be a member of more than one community.

A system (or part of it) and the hardware facilities which *active roles* interact with are modelled as *passive roles.*

3.1.2.1 Roles in the Data Acquisition Community

The main objectives of the data acquisition community is to bring measurements into the system. The measurement and monitoring models are designed by *model designer* based on the requirements of *environmental scientists*. Such a design decides what data is to be collected and what metadata is to be associated with it, such as experimental information and instrument conditions. *Technicians* configure and calibrate a *sensor* or a *sensor network* to satisfy the experiment specifications. In the case where human sensors are to be used, *observers* or *measurers* input the measures to the system, e.g., by using mobile devices. *Data collectors* interact with a *data acquisition system* to prepare the data or control the flow of data and automatically collect and transmit the data.

As shown in Figure 4.2, the following roles are identified in the data acquisition community:

- **Environmental Scientist**: An active role, which is a person who conducts research or performs investigation for the purpose of identifying, abating, or eliminating sources of pollutants or hazards that affect either the environment or the health of the population. Using knowledge of

various scientific disciplines, they may collect, synthesize, study, report, and recommend action based on data derived from measurements or observations of air, food, soil, water, and other sources.

- **(Measurement Model) Designer**: An active role, which is a person who designs the measurements and monitoring models based on the requirements of environmental scientists.
- **Sensor**: A passive role, which is a converter that measures a physical quantity and converts it into a signal which can be read by an observer or by an (electronic) instrument.
- **Sensor network:** A passive role, which is a network consisting of distributed autonomous sensors to monitor physical or environmental conditions.
- **Technician**: An active role, which is a person who develops and deploys sensor instruments, establishing and testing the sensor network, operating, maintaining, monitoring and repairing the observatory hardware.
- **Measurer**: An active role, which is a person who determines the ratio of a physical quantity, such as a length, time, temperature etc., to a unit of measurement, such as the meter, second or degree Celsius.
- **Observer**: An active role, which is a person who receives knowledge of the outside world through the senses, or records data using scientific instruments.
- **Data collector**: An active role, which is a person who prepares and collects data. The purpose of data collection is to obtain information to keep on record, to make decisions about important issues, or to pass information on to others.
- **Data Acquisition Subsystem**: In Science Viewpoint, data acquisition subsystem represents a passive role of the data acquisition community. As defined in Section 2 Model Overview, it is research infrastructure which provides functionalities to automate the process of data acquisition.

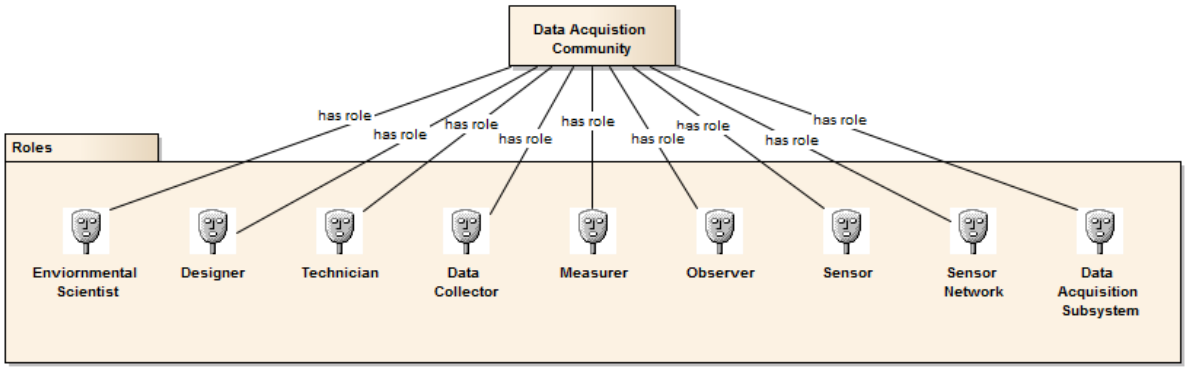


Figure 4.2: Roles in the Data Acquisition Community

3.1.2.2 Roles in the Data Curation Community

The data curation community responds to provide quality data products and maintain the data resources. Consider a typical data curation scenario: when data is being imported into a curation subsystem, a curator will perform the quality checking of the scientific data. Unique identifiers will be assigned to the qualified data, which will then be properly catalogued by associating necessary metadata, and stored or archived. The main human roles interacting with or maintaining a data curation subsystem are data curators who manage the data and storage administrators who manage the storage facilities.

As shown in Figure 4.3, we identified the following roles in this community:

- **Data Curator**: An active role, which is a person who verifies the quality of the data, preserves and maintain the data as a resource, and prepares various required data products.
- **Data Curation System**: In Science Viewpoint, data curation subsystem represents a passive role of the data curation community. The definition is given in Section 2 Model Overview, which a research infrastructure stores, manages and ensures access to all persistent data-sets produced within the infrastructure.
- **Storage Administrator**: An active role, which is a person who has the responsibilities to design data storage, tune queries, perform backup and recovery operations, set up RAID mirrored arrays, and make sure drive space is available for the network.
- **Storage**: A passive role, which includes memory, components, devices and media that retain digital computer data used for computing for some interval of time.

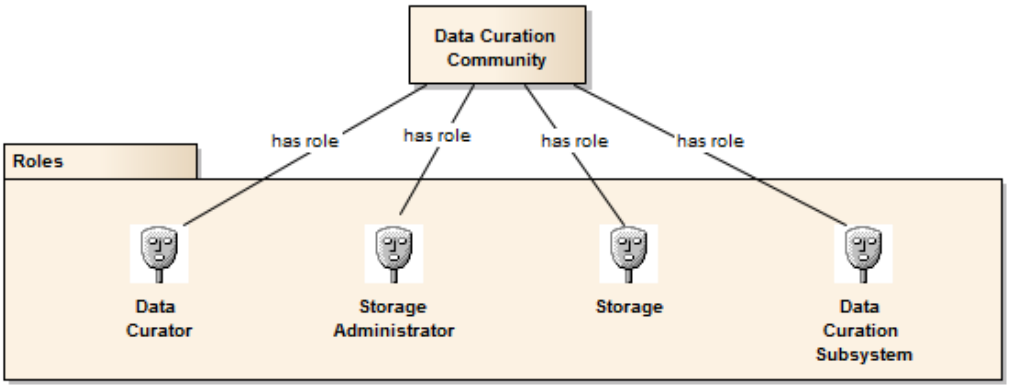


Figure 4.3: Roles in the Data Curation Community

3.1.2.3 Roles in the Data Publication Community

The objectives of the data publication community are to publish data and assist discovery and access. We consider the scenarios described by Kahn's data publication model [34]: an *originator*, i.e., a user with digital material to be made available for public access, makes the material into a digital object. A digital object is a data structure whose principal components are digital material, or data, plus a unique identifier for this material, called a *handle* (and, perhaps, other material). To get a *handle*, the user requests one from an authorized *handle generator*. A user may then deposit the digital object in one or more *repositories*, from which it may be made available to others (subject, to the particular item's terms and conditions, etc.). Upon depositing a digital object in a repository, its *handle* and the *repository* name or IP address is registered with a globally available system of *handle servers*. Users may subsequently present a *handle* to a *handle server* to learn the network names or addresses of repositories in which the corresponding digital object is stored. We use a more general term "PID" instead of "*handle"* (thus, "*PID registry*" instead of "*handle servers*"), and identify the key roles involved in the data publication process including, a data originator, a PID generator, a repository, and a PID registry.

The published data are to be discovered and accessed by data consumers. A semantic mediator is used to facilitate the heterogeneous data discovery.

In summary, as shown in Figure 4.4, the following roles are involved in the data publication community:

- **Data Originator**: Either an active or a passive role, which provides the digital material to be made available for public access.
- **PID Generator**: A passive role, a system which assigns persist global unique identifiers to a (set of) digital object.
- **PID Registry**: A passive role, which is an information system for registering PIDs.
- **(Data Publication) Repository**: A passive role, which is a facility for the deposition of published data.
- **Semantic Mediator**: A passive role, which is a system or middleware facilitating semantic mapping discovery and integration of heterogeneous data.
- **Data Access Subsystem**: In Science Viewpoint, data access subsystem represents a passive role of the data publication community. The definition is given in Section 2 Model Overview, which is a research infrastructure enables discovery and retrieval of scientific data subject to authorisation.
- **Data Consumer**: Either an active or a passive role, which is an entity who receives and uses the data.

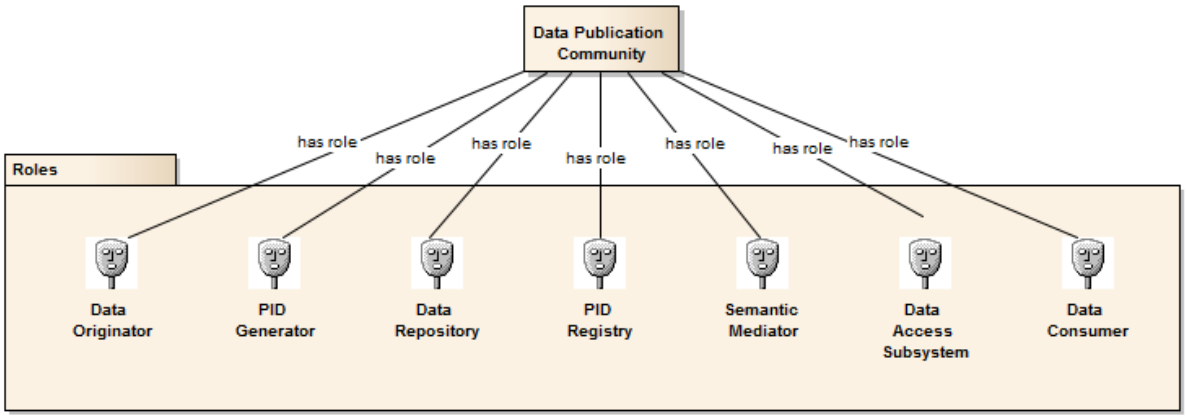


Figure 4.4: Roles in the Data Publication Community

3.1.2.4 Roles in the Data Service Provision Community

The data service provision community provides various application services such as data analysis, mining, simulation and modelling, visualisation, and experimental software tools, in order to facilitate the use of the data. We consider scenarios of service oriented computing paradigm which is adopted by the ENVRI implementation model, and identify the key roles as below. These concepts are along the lines of the existing standards such as OASIS Reference Model for Service Oriented Architecture.

As shown in Figure 4.5, roles in the data service provision community include:

- **Data Provider**: Either an active or a passive role, which is an entity providing the data to be used.
- **Service Provider**: Either an active or a passive role, which is an entity providing the services to be used.
- **Service Registry**: A passive role, which is an information system for registering services.
- **Capacity Manager**: An active role, which is a person who manages and ensures that the IT capacity meets current and future business requirements in a cost-effective manner.

SEVENTH FRAMEWORK PROGRAMME

- **Service Consumer**: Either an active or a passive role, which is an entity using the services provided.

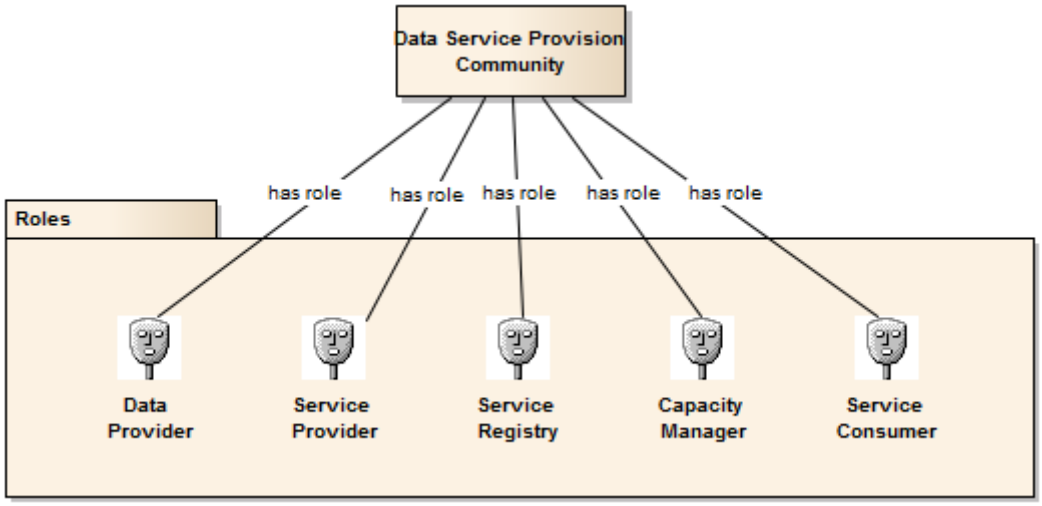


Figure 4.5: Roles in the Data Service Provision Community

3.1.2.5 Roles in the Data Usage Community

The main role in the data usage community is a *user* who is the ultimate consumer of data, applications and services. Depending on the purposes of usage, a user can be one of the following active roles:

- **Scientist or Researcher**: An active role, which is a person who makes use of the data and application services to conduct scientific research.
- **Technologist or Engineer**: An active role, which is a person who develops and maintains the research infrastructure.
- **Education or Trainee**: An active role, which is a person who makes use of the data and application services for education and training purposes.
- **Policy or Decision Maker**: An active role, which is a person who makes decisions based on the data evidences.
- **Private Sector (Industry investor or consultant)**: An active role, which is a person who makes use of the data and application service for predicting markets so as to make business decisions on producing related commercial products.
- **General Public, Media or Citizen (Scientist)**: An active role, which is a person who is interested in understanding the knowledge delivered by an environmental science research infrastructure, or discovering and exploring the knowledgebase enabled by the research infrastructure.

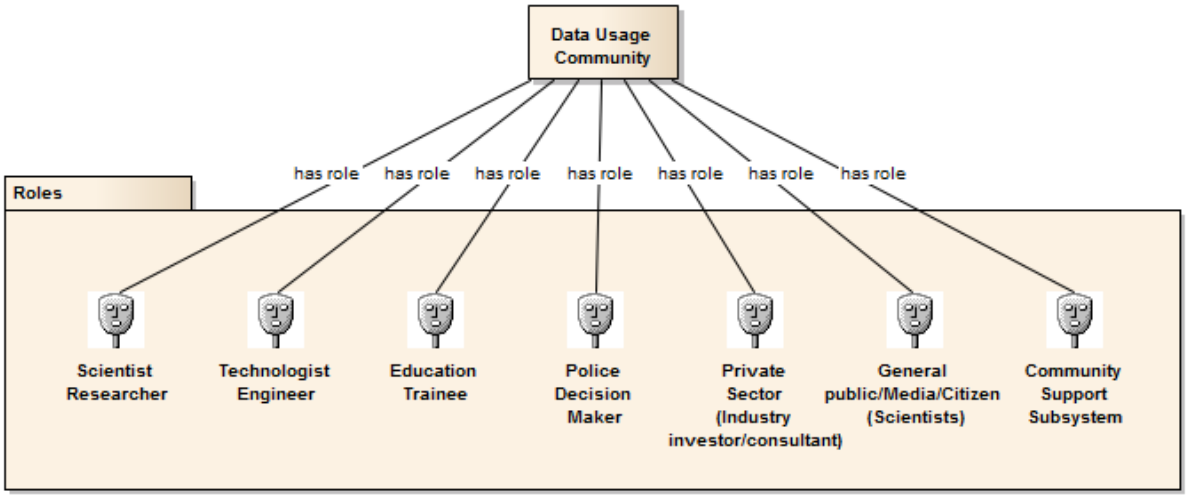


Figure 4.6: Roles in the Data Usage Community

3.1.3 Common Community Behaviours

A **behaviour** of a community is a composition of actions performed by roles normally addressing separate business requirements. In the ENVRI-RM, the modelling of community behaviours is based on analysis of the common requirements of the ENVRI research infrastructure which has resulted in a list of common functions. The initial model focuses on the minimal set of requirements. A community behaviour can be either a single function or a composition of several functions from the function list.

3.1.3.1 Behaviours of the Data Acquisition Community

Figure 4.7 depicts the main behaviours of the data acquisition community including:

- **Design of Measurement Model**: A behaviour performed by a *Measurement Model Designer* that designs the measurement or monitoring model based on scientific requirements.
- **Instrument Configuration**: A behaviour performed by a *Technician* that sets up a *sensor* or a *sensor network*.
- **Instrument Calibration**: A behaviour performed by a *Technician* that controls and records the process of aligning or testing a *sensor* against dependable standards or specified verification processes.
- **Data Collection**: A behaviour performed by a *Data Collector* that obtains digital values from a *sensor* instrument (or a human sensor such as a *Measurer* or an *Observer*), associating consistent timestamps and necessary metadata

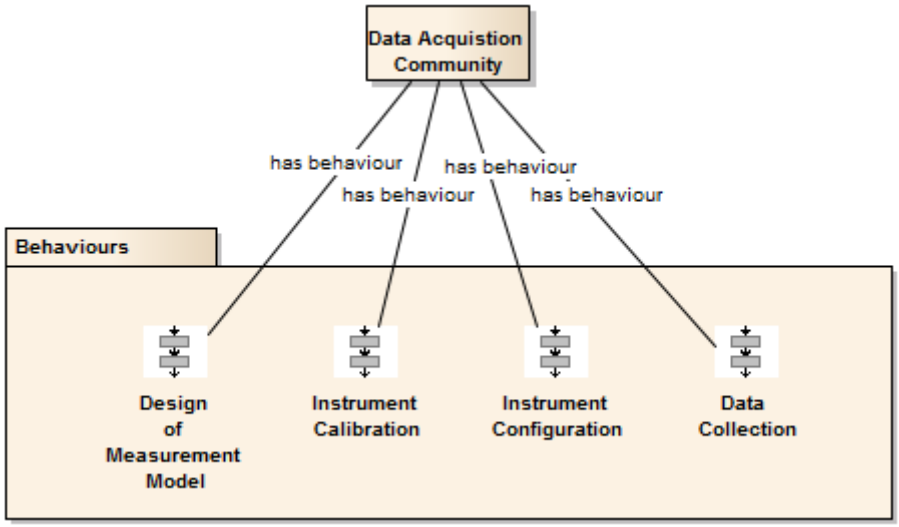


Figure 4.7: Behaviours of the Data Acquisition Community

3.1.3.2 Behaviours of the Data Curation Community

The main behaviours of the data curation community are depicted in Figure 4.8 which include:

- **Data Quality Checking**: A behaviour performed by a *Data Curator* that detects and corrects (or removes) corrupt, inconsistent or inaccurate records from data sets.
- **Data Preservation**: A behaviour performed by a *Data Curator* that deposits (over long-term) the data and metadata or other supplementary data and methods according to specified policies, and makes them accessible on request.
- **Data Product Generation**: A behaviour performed by a *Data Curator* that processes data against requirement specifications and standardised formats and descriptions.
- **Data Replication**: A behaviour performed by a *Storage Administrator* that creates, deletes and maintains the consistency of copies of a data set on multiple storage devices.

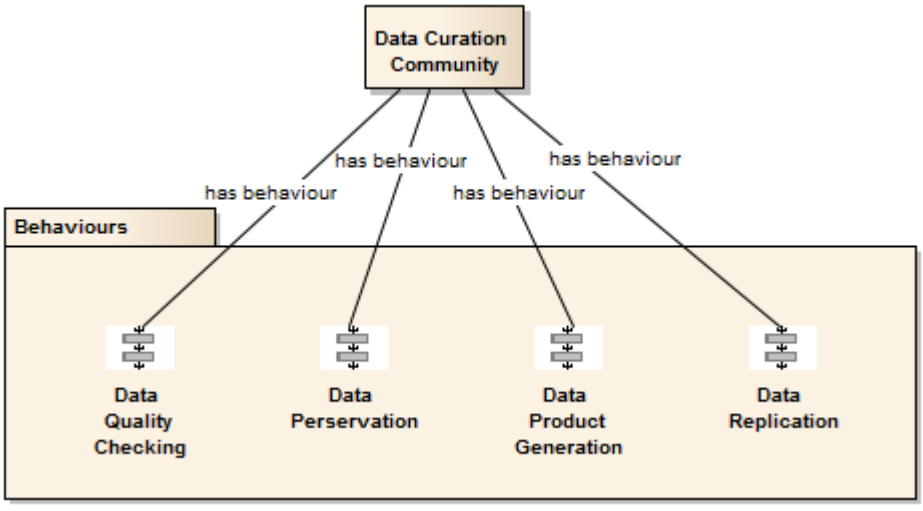


Figure 4.8: Behaviours of the Data Curation Community

3.1.3.3 Behaviours of the Data Publication Community

As shown in Figure 4.9, a data publication community may perform the following behaviours:

- **Data Publication**: A behaviour that provides clean, well-annotated, anonymity-preserving datasets in a suitable format, and by following specified data-publication and sharing policies, to make the datasets accessible publicly or to those who agree to certain conditions of use, and to individuals who meet certain professional criteria.
- **Semantic Harmonisation**: A behaviour enabled by a *Semantic Mediator* that unifies similar data (knowledge) models based on the consensus of collaborative domain experts to achieve better data (knowledge) reuse and semantic interoperability.
- **Data Discovery and Access**: A behaviour enabled by a *Data Discovery and Access system* that retrieves requested data from a data resource by using suitable search technology.
- **Data Citation**: A behaviour performed by a *Data Consumer* that assigns an accurate, consistent and standardised reference to a data object, which can be cited in scientific publications.

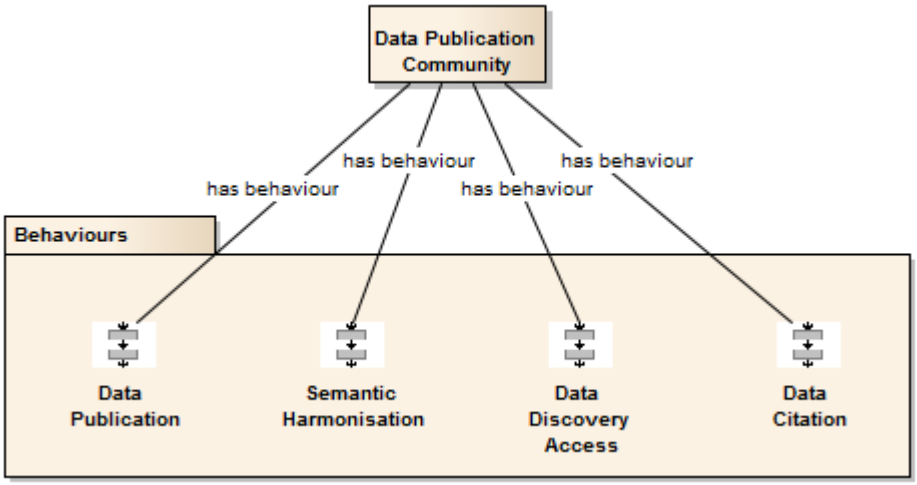


Figure 4.9: Behaviours of the Data Publication Community

3.1.3.4 Behaviours of the Data Service Provision Community

Figure 4.10 depicts the behaviours modelled for the data service provision community, which inclued:

- **Service Description**: A behaviour performed by a *Service Provider* to provide the information needed in order to use a service [8].
- **Service Registration**: A behaviour performed by a *Service Provider* to make the service visible to *Service Consumers* by registering it in a service registry [8].
- **Service Coordination**: A behaviour performed by a *Service Provider* to coordinate the actions of distributed applications in order to reach consistent agreement on the outcome of distributed transactions.
- **Service Composition**: A behaviour performed by a *Service Provider* to combine multiple services which can be achieved by either *Choreography* or *Orchestration*. ***Service Choreography*** is a collaboration between *Service Providers* and *Service Consumers*. ***Service Orchestration*** is the behaviour that a *Service Provider* performs internally to realise a service that it provides [35].

These are general behaviours of a service-oriented computing model. In the context of environmental science research infrastructures, a data service provision community will focus on the implementation of domain special services, in particular those supporting **Data Assimilation, Data Analysis**, **Data Mining, Data Extraction, Scientific Modelling and Simulation, (Scientific) Workflow Enactment. (**See Chapter 2, Terminology and Glossary**,** for the definitions of these functionalities.)

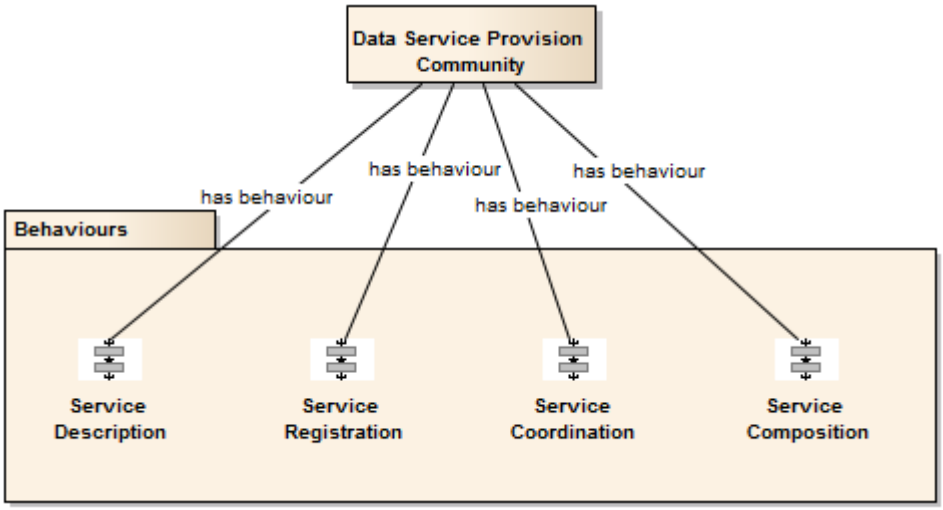


Figure 4.10: Behaviours of the Data Service Provision Community

3.1.3.5 Behaviours of the Data Usage Community

Finally, a data usage community may have the following behaviours, which is depicted in Figure 4.11:

- **User Behaviour Tracking**: A behaviour enabled by a *Community Support System* to track the *Users*. If the research infrastructure has identity management, authorisation mechanisms, accounting mechanisms, for example, a Data Access Sub-System is provided, then the Community Support System either include these or work well with them.
- **User Profile Management**: A behaviour enabled by a *Community Support System* to support persistent and mobile profiles, where profiles will include preferred interaction settings, preferred computational resource settings, and so on.
- **User Working Space Management**: A behaviour enabled by a *Community Support System* to support work spaces that allow data, document and code continuity between connection sessions and accessible from multiple sites or mobile smart devices.
- **User Working Relationships Management**: A behaviour enabled by a *Community Support System* to support a record of working relationships, (virtual) group memberships and friends.
- **User Group Work Supporting**: A behaviour enabled by a *Community Support System* to support controlled sharing, collaborative work and publication of results, with persistent and externally citable PIDs.

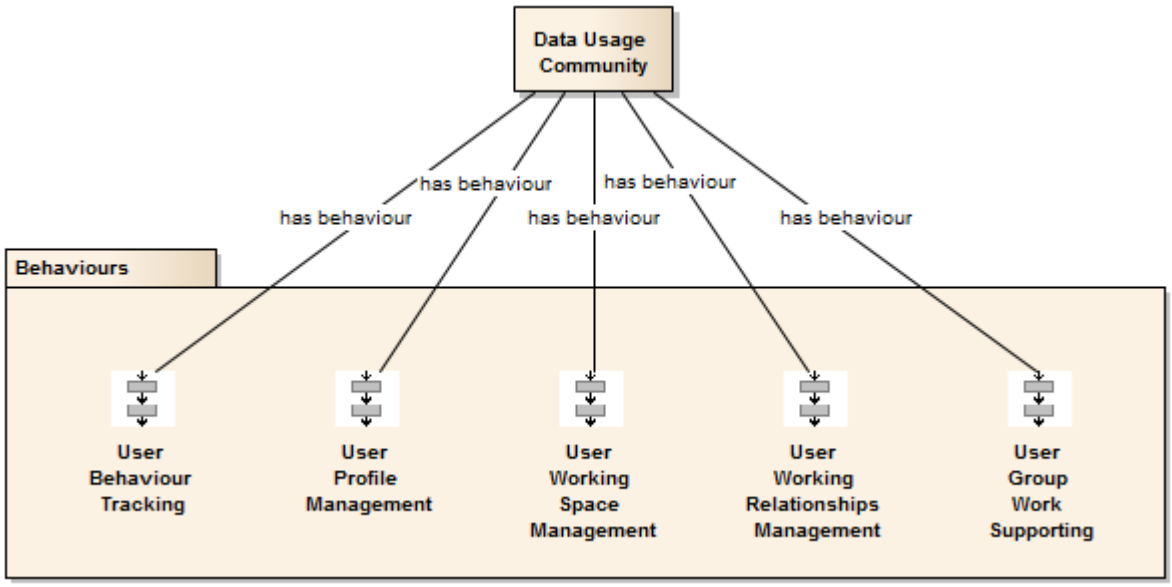


Figure 4.11: Behaviours of the Data Usage Community

3.2 Information Viewpoint

The goal of the information viewpoint is to provide a common abstract model for the shared information handled by the infrastructure. The focus lies on the information itself, without considering any platform-specific or implementation details. It is independent from the computational interfaces and functions that manipulate the information or the nature of technology used to store it. Similar to a high level ontology, it aims to provide a unique and consistent interpretation of the shared information entities of a particular domain. The information viewpoint specifies the types of the information objects and the relationships between those types. It describes how the state of the data evolves as the results of computational operations. It also defines the constraints on the data and the rules governing the processing of such data.

In the information viewpoint, we discuss the following aspects:

- **Components**, which defines a collections of information objects and action types necessary to support the minimal set of required functionalities (See Section 2 Model Overview);
- **Dynamic Schemata**, which specifies how the information objects evolve as the system operates, describing the allowable state changes as the effects of the actions;
- **Static Schemata**, which describes instantaneous views of the information objects at a certain stage of the data lifecycle; and
- **Subsystems**, which regroups the defined information objects into the 5 common Subsystems (as defined in Section 2 Model Overview) for the purpose of easy observation.

3.2.1 Components

The ENVRI information specification defines a configuration of information objects, the behaviour of those objects, the actions that can happen and a set of constraints that should always hold for this collection of elements. The model elements are organised into four groups:

- Information Objects, which defines a collection of information objects manipulated by the system;

SEVENTH FRAMEWORK PROGRAMME

- Information Action Types, which defines events that cause state changes of information objects;
- Information Object Instances, which specifies realised variations of defined information objects;
- Data States, which defines data states and their changes as effects of actions.

3.2.1.1 Information Objects

Information objects are used to model various information entities manipulated by the system. In ENVRI, information objects are defined to capture 3 types of information:

- The meta information of data collections, typically those related to the design of observation and measurement models, including:
  - the design specification of the observation and measurement;
  - the description of the measurement procedure;
- The data or information processed by the system, mainly linked with the persisted data, including: the scientific data;
  - the Quality Assurance (QA) annotations;
  - the metadata or concepts from a conceptual model e.g., an ontology;
  - the unique identifiers for the data identification;
  - the various data states as the effects of actions;
- The information used for the management of data, including:
  - the backup (of data)
  - the mapping rules which are used for the model-to-model transformations; and
  - the data provenance which are used to record the state changes of data in their lifecycles

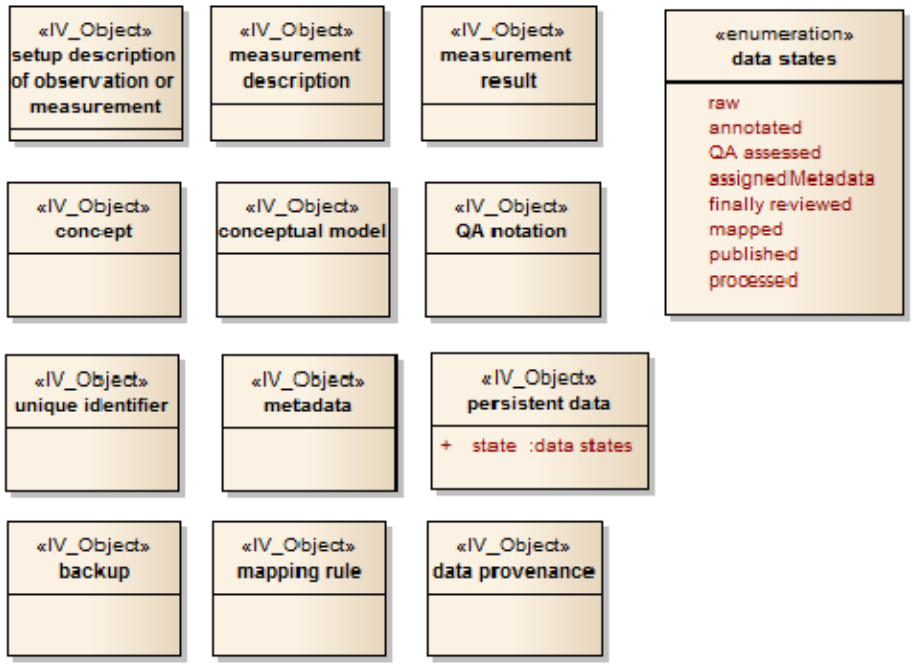


Figure 4.12: Information Objects

The definitions of the information objects given in Figure 4.12 are given as follows:

specification of scientific measurements

The description of the scientific measurement model which specifies:

- what is measured;
- how it is measured;
- by whom it is measured; and
- what the temporal design is (single /multiple measurements / interval of measurement etc.)

Note, the specification of measurement can be included as metadata or the semantic annotations of the scientific data to be collected. It is important such a design specification is both explicit and correct, so as to be understood or interpreted by external users or software tools. Ideally, a machine readable specification is desired.

measurement result

Quantitative determinations of magnitude, dimension and uncertainty to the outputs of observation instruments, sensors (including human observers) and sensor networks.

QA notation

Notation of the result of a Quality Assessment. This notation can be a nominal value out of a classification system up to a comprehensive (machine readable) description of the whole QA process.

In practices, this can be:

- simple flags like "valid" / "invalid" up to comprehensive descriptions like
- "data set to invalid by xxxxxx on ddmmyy because of yyyyyyy"

QA notation can be seen as a special annotation. To allow sharing with other users, the QA notation should be unambiguously described so as to be understood by others or interpretable by software tools.

unique identifier (UID)

With reference to a given (possibly implicit) set of objects, a unique identifier (UID) is any identifier which is guaranteed to be unique among all identifiers used for those objects and for a specific purpose.

There are 3 main generation strategy:

- serial numbers, assigned incrementally;
- random numbers, selected from a number space much larger than the maximum (or expected) number of objects to be identified. Although not really unique, some identifiers of this type may be appropriate for identifying objects in many practical applications and are, with abuse of language, still referred to as "unique";
- names or codes allocated by choice which are forced to be unique by keeping a central registry.

The above methods can be combined, hierarchically or singly, to create other generation schemes which guarantee uniqueness.

In many cases, a single object may have more than one unique identifier, each of which identifies it for a different purpose. For example, a single object can be assigned with the following identifiers:

- global: unique for a higher level community
- local: unique for the subcommunity

The critical issues of unique identifiers include but not limited to:

- long term persistence – without efficient management tools, UIDs can be lost;
- resolvability -- without efficient management tools, the linkage between a UID and its associated contents can be lost.

metadata

Data about data, in scientific applications is used to describe, explain, locate, or make it easier to retrieve, use, or manage an information resource.

There have been numerous attempts to classify the various types of metadata. As one example, NISO (National Information Standards Organisation) distinguishes between three types of metadata based on their functionality: Descriptive metadata, which describes a resource for purposes, such as discovery and identification; Structural metadata, which indicates how compound objects are put together; and Administrative metadata, which provides information to help manage a resource. But this is not restrictive. Different applications may have different ways to classify their own metadata.

Metadata is generally encoded in a metadata schema which defines a set of metadata elements and the rules governing the use of metadata elements to describe a resource. The characteristics of metadata schema normally include: the number of elements, the name of each element, and the meaning of each element. The definition or meaning of the elements is the semantics of the schema, typically the descriptions of the location, physical attributes, type (i.e., text or image, map or model), and form (i.e., print copy, electronic file). The value of each metadata element is the content. Sometimes there are content rules and syntax rules. The content rules specify how content should be formulated, representation constraints for content, allowable content values and so on. And the syntax rules specify how the elements and their content should be encoded. Some popular syntax used in scientific applications include Some popular syntax includes:

- HTML (Hyper-Text Markup Language): www.w3.org/MarkUp/
- XML (eXtensible Markup Language): www.w3.org/XML/
- RDF (Resource Description Framework): www.w3.org/RDF/
- OWL (Web Ontology Language): www.w3.org/2001/sw/
- SGML (Standard Generalised Markup Language): www.w3.org/MarkUp/SGML/
- MARC (Machine Readable Cataloging): www.loc.gov/marc/
- MIME (Multipurpose Internet Mail Extensions): www.ukoln.ac.uk/metadata/resources/mime/
- DIME(Direct Internet Message Encapsulation): xml.coverpages.org/draft-nielsen-dime-01.txt

Such syntax encoding allows the metadata to be processed by a computer program.

Many standards for representing scientific metadata have been developed within disciplines, sub-disciplines or individual project or experiments. Some widely used scientific metadata standards include:

- Dublin Core: purl.oclc.org/metadata/dublin_core/
- ISO 11179: metadata-stds.org/11179/
- FGDC (The Federal Geographic Data Committee): www.fgdc.gov/standards
- DDI (Data Documentation Initiative): www.ddialliance.org/
- INSPIRE: http://inspire.jrc.ec.europa.eu/
- TEI (The Text Encoding Initiative): www.tei-c.org/
- METS (Metadata Encoding and Transmission Standard): www.loc.gov/standards/mets/
- MODS (Metadata Object Description Schema): www.loc.gov/standards/mods/
- OAIS (Reference Model for an Open Archival Information System)

Two aspects of metadata give rise to the complexity in management:

- Metadata are data, and data become metadata when they are used to describe other data. The transition happens under particular circumstances, for particular purposes, and with certain perspectives, as no data are always metadata. The set of circumstances, purposes, or perspectives for which some data are used as metadata is called the 'context'. So metadata are data about data in some 'context'.
- Metadata can be layered. This happens as data objects or information resources may move to different phases during their life in a digital environment, thus requiring layers of metadata that can be associated.

Metadata can be fused with the data. However, in many applications, such as a provenance system which tracks the environmental experience workflows, or a distributed satellite image annotation system the metadata and data, metadata and data are often created and stored separately, e.g., they may be generated by different users, in different computing processes, stored at different locations, in different types of storage. Metaata and data may have different lifecycles, e.g., they might have been created at a different time, updated and removed independently. Often, there is more than one set of metadata related to a single data resource, e.g., when the existing metadata becomes insufficient, users may design new templates to make another metadata collection. Without efficient software and tools, the management of the linkage between metadata and data becomes onerous. Such linkage relationship between metadata and data are vulnerable to failures in the processes that create and maintain them, and to failures in the systems that store their representations. It is important to devise methods that reduce these failures.

concept

Name and definition of the meaning of a thing (abstract or real thing). Human readable definition by sentences, machine readable definition by relations to other concepts (machine readable sentences). It can also be meant for the smallest entity of a conceptual model. It can be part of a flat list of concepts, a hierarchical list of concepts, a hierarchical thesaurus or an ontology.

conceptual model

A collection of concepts, their attributes and their relations. It can be unstructured or structured (e.g. glossary, thesaurus, ontology). Usually the description of a concept and/or a relation defines the concept in a human readable form. Concepts within ontologies and their relations can be seen as machine readable sentences. Those sentences can be used to establish a self description. It is, however, practice today, to have both, the human readable description and the machine readable description. In this sense a conceptual model can also be seen as a collection of human and machine readable sentences. Conceptual models can reside within the persistence layer of a data provider or a community or outside. Conceptual models can be fused with the data (e.g. within a network of triple stores) or kept separately.

data state

Term used as defined in ISO/IEC 10746-2. At a given instant in time, data state is the condition of an object that determines the set of all sequences of actions (or traces) in which the object can participate.

The data states and their changes as effects of actions are specified in subsection 3.2.1.4 Data States.

These states are referential states. The instantiated chain of data lifecyle can be expressed in data provenance.

(persistent) data

Term used as defined in ISO/IEC 10746-2. Data is the representations of information dealt by information systems and users thereof.

backup

A copy of computer data so it may be used to restore the original after a data loss event.

mapping rule

Configuration directives used for model-to-model transformation. They can be:

- transformation rules for arithmetic values (mapping from one unit to another) from linear functions like k.x + d to multivariate functions
- transformation rules for ordinal and nominal values e.g. transforming classifications according to a classification system A to classification system B
- transformation rules for data descriptions (metadata or semantic annotation or QA annotation)
- transformation rules for Parameter names and descriptions (can be n:m)
- transformation rules for Method names and descriptions
- transformation rules for Sampling descriptions

data provenance

Information that traces the origins of data and records all state changes of data during their lifecycle and their movements between storages.

An creation of an entrance into the data provenance records triggered by any actions typically contain:

- date of action;
- actor;

- type of action;
- data_id.

Data provenance system is an annotation system for managing data provenances. Usually unique identifiers are used to refer the data in their different states and for the description of the different states.

3.2.1.2 Information Action Types

Information actions model the information processing in the system. Every action is associated with at least one object. Actions cause state changes in the objects that participate in them.

Figure 4.13 shows a collection of all action types used to model the information viewpoint.

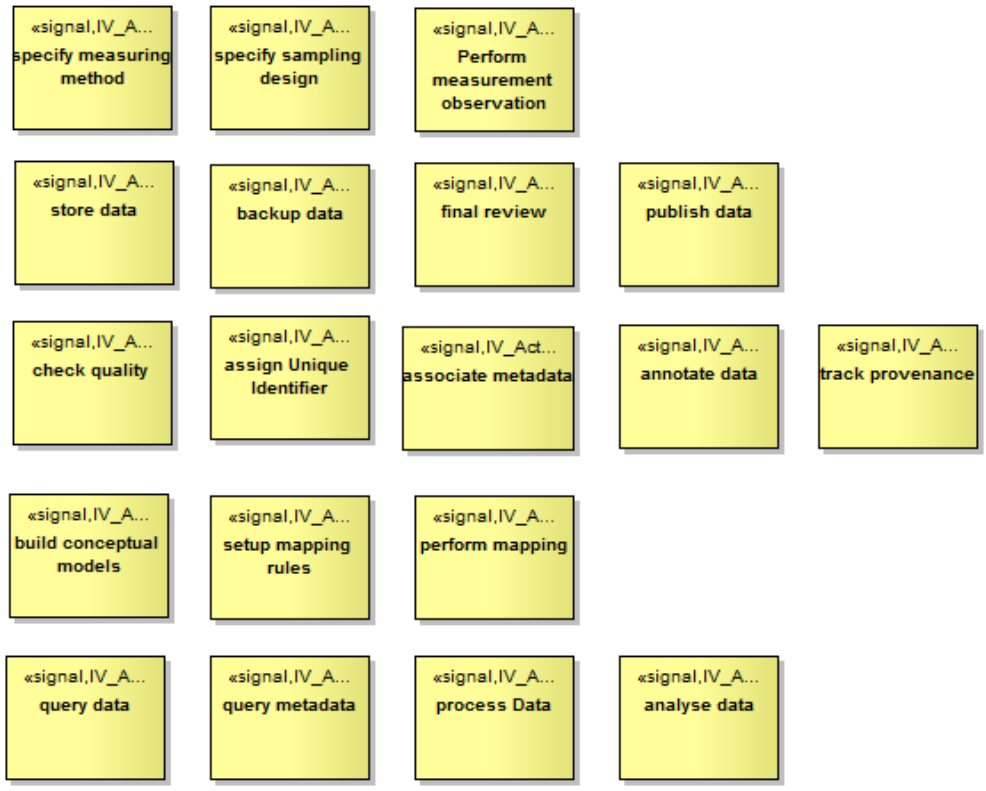


Figure 4.13: Information Objects Action Types

specify measuring method

Specify the details of the method of observations/measurements.

For example, this may include the specification of a measurement device type and its settings, measurement/ observation intervals.

specify sampling design

Specify design of sampling or observation.

For example this may include:

- geographical position of measurement or observation (site) -- the selections of observations and measurement sites, e.g., can be statistical or stratified by domain knowledge;
- characteristics of site;
- preconditions of measurements.

perform measurement observation

Measure parameter(s) or observe an event. The performance of a measurement or observation produces measurement results.

store data

Archive or preserve data in persistent manner to ensure continuing accessible and usable.

backup data

Replicate data to an additional data storage so it may be used to restore the original after a data loss event. A special type of backup is a long term preservation.

final review

Review the data to be published, which will not likely be changed again.

The action triggers the change of the data state to be "finally reviewed". In practices, an annotation for such a state change should be recorded for provenance purposes. Usually, this is coupled with archiving and versioning actions.

publish data

Make data public accessible.

For example, this can be done by:

- presenting them in browsable form on the world wide web
- by presenting them via special services:
    - RESTful service
    - SOAP service
    - OPEN GRID service
    - OGC service (web feature service, web map service)
    - SPARQL endpoint

check quality

Actions to verify the quality of data.

For example it may involve:

- remove noise
- remove apparently wrong data
- calculate calibrations

Quality checks can be carried out at different points in the chain of data lifecycle.

Quality checks can be supported by software tools for those processes which can be automated (e.g. statistic tolerance checks).

assign Unique Identifier

Obtain a unique identifier and associate it to the data.

associate metadata

Add additional information according to a predefined schema (metadata schema). This partially overlaps with data annotations.

annotate data

Annotate data with meaning (concepts of predefined local or global conceptual models).

In practices, this can be done by adding tags or a pointer to concepts within a conceptual model to the data. If the concepts are terms e.g., in an SKOS/RDF thesaurus, and published as linked data, then data annotation would mean to enter the URL of the term describing the meaning of the data. There is no exact borderline between metadata and semantic annotation.

track provenance

Add information about the actions and the data state changes as data provenances.

build conceptual models

Create a local or global model of interrelated concepts.

This may involve the following issues:

- commitment: the agreement of a larger group of scientists /data providers / data users should be achieved;
- unambiguousness: the concept model should be unambiguously defined;
- readability: the model should be readably by both human and machine. E.g., ontologies should express the meaning of the concepts with the relations to other concepts. This form is human and machine readable. Recently it has increasingly become important to add definitions in human readable language.
- availability: the conceptual model must be referenceable and dereferenceable for a long time

setup mapping rules

Specify the mapping rules of data and/or concepts.

These rules should be explicitly expressed by a language so that can be processed by software tools.

A minimal set of mapping rule should include the following information:

- source data / concept for which the mapping is valid
- target data / concept, for which the mapping is valid
- mapping process (the translation and or transformation process)
- validity constraints for the mapping (temporal constraints, context constraints, etc.)

SEVENTH FRAMEWORK PROGRAMME

perform mapping

Execute transformation rules for values (mapping from one unit to another unit) or translation rules for concepts (translating the meaning from one conceptual model to another conceptual model, e.g. translating code lists).

query data

Send a request to data resources to retrieve data of interests.

In practices, there are two types of data query exist:

- two step approach:
  step 1: query/search metadata;
  step 2: access data

For example, when using OGC services, it usually first invokes a web feature service to obtain feature descriptions, then a web map service can be invoked to obtain map images.

- one step approach: to query data e.g., by using SQL services or SPARQL endpoints

Requests can be directly sent to a service or distributed by a broker.

query metadata

Send a request to metadata resources to retrieve metadata of interests.

process data

Process data for the purposes of:

- converting and generating data products
- calculations: e.g., statistical processes, simulation models
- visualisation: e.g., alpha-numerically, graphically, geographically

Data processes should be recorded as provenance

analyse data

Perform a sequence of executions of metadata/data querying→ data processing→ result interpreting→ a new query, in order to deepen the knowledge about the data.

It can be supported by software tools, e.g., a workflow engine, that helps to carry out that sequence of data request and interpretation of results.

3.2.1.3 Information Object Instances

Figure 4.14 shows the collection of instances of information objects which are information objects existing more than once having several instances.

Instances of information objects are needed for two purposes:

1. to show the data state changes as effects of actions;
2. to show the fact that there usually is a "local conceptual model" and a "global conceptual model", both of which are conceptual models.

global concept

Concepts with a commitment of a whole data sharing community. Usually those concepts are part of global conceptual models (global Thesauri like GEMET / EuroVoc / AGROVOC or global ontologies like Gene Ontology , ...).

local concept

A concept of

- person
- institute
- anything else

A concept can be local or global depending on the size of the community which commits to it and only if considered in relation to each other.

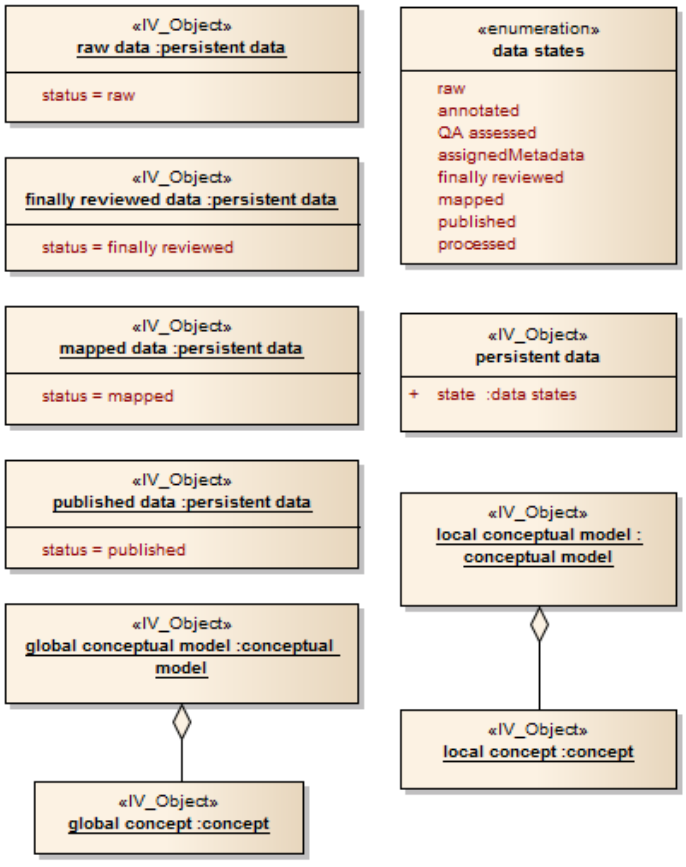


Figure 4.14: Instances of Information Objects

3.2.1.4 Data States

In the ENVRI Information specification, the following data states are defined:

- **raw**: the primary results of observations or measurements;
- **annotated**: data that are connected to concepts, describing their meaning;

- **QA assessed**: data that have undergone checks and are connected with descriptions of the results of those checks;
- **metadata associated**: data that are connected to metadata which describe those data;
- **finally reviewed**: data that have undergone a final review and therefore will not be changed anymore;
- **mapped**: data that are mapped to a certain conceptual model;
- **published**: data that are presented to the outside world;
- **processed**: data that have undergone a processing (evaluation, transformation).

As an example, the state changes as effects of actions are illustrated in Figure 4.15.

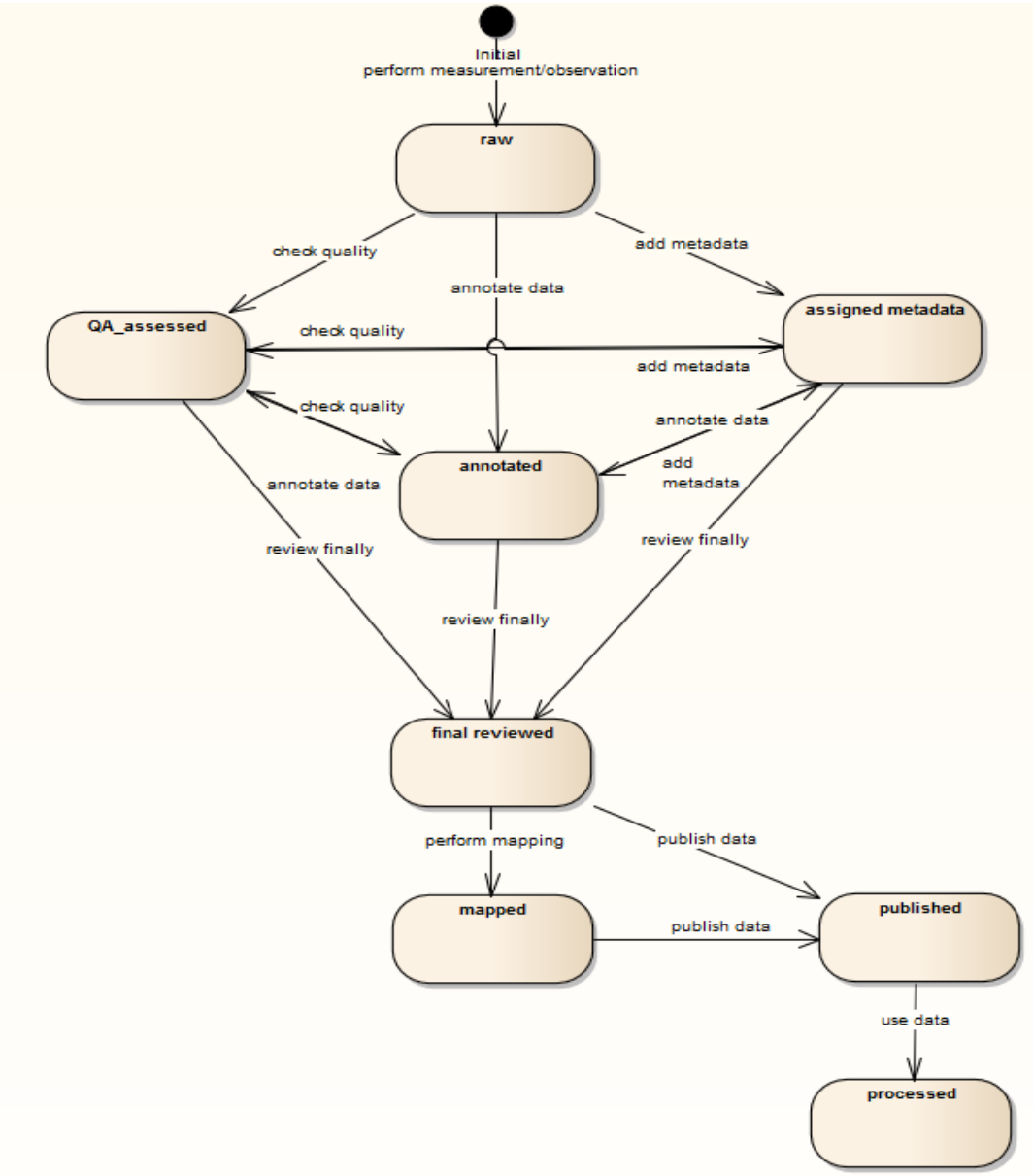


Figure 4.15: An Example of Data States Changes As Effects of Actions

3.2.2 Dynamic Schemata

In information viewpoint, dynamic schemata specify how the information evolves as the system operates, describing the allowable state changes of one or more information objects [37].

The specification consists of 2 parts:

- **Dynamic Schemata Overview**, which is an overview of the schemata and state changes of information objects as effects of actions;
- **Tracing of Provenance**, which summarises the provenance tracking points.

3.2.2.1 Dynamic Schemata Overview

An overview of the specification is illustrated in Figure 4.16. A more detailed specification is given in Appendix C.

Before a measurement or observation can be started the design (or setup) must be defined, including the working hypothesis and scientific question, method of the selection of sites (stratified / random), necessary precision of the observation or measurement, boundary conditions, etc. For correctly using the resulting data, detailed information about that process and its parameters have to be available for people processing the data. (e.g. if a stratified selection of sites according to parameter A is done, the resulting value of parameter A cannot be evaluated in the same way as other results)

After defining the overall design of measurements or observations, the measurement method, complying with the design, including devices which should be used, standards / protocols which should be followed, and other details have to be specified. Information of that process and the parameters resulting of the process have to be stored in order to guarantee correct interpretation of the resulting data. (e.g. when you want to model a dependency of parameter B of a parallel measured wind velocity, the limit of detection of the used anemometer influences the range of values of possible assertions).

When the measurement or observation method is defined, it can be carried out, producing measurement results. The handling of those results, all the actions done, to store the data are pulled together in the action "store data". (This action can be very simple when using a measurement device, which periodically sends the data to the data management system, but this can also be a sophisticated harvesting process or e.g. in case of biodiversity observations a process done by humans). The storage process is the first step in the life cycle of data that makes data accessible in digital form and are persisted.

As soon as data are available for IT purposes a backup can be made, independently of the state of the persisted data. This can be done locally or remote, done by the data owners or by dedicated data curation centers. At any status of the data data can be processed for QA-assessments, for readjustment of the measurement or observation design and a lot of other reasons. Evaluations, which lead to answers of the scientific question, however, are usually done on data with a certain status - the status "finally reviewed".

Making data accessible for users outside the Environment of the data owner at least needs two steps: 1) Mapping the data to the "global" semantics, the semantics the data owner shares with the data user. 2) Publish the data. Mapping data to global semantics may include simple conversions like

conversions of units but also need more sophisticated transformations like transformations of code lists and other descriptions like the setup descriptions, measurement descriptions, and data provenance.

It is important to know about published data, whether those data have a status: "finally reviewed" and what such a status means. It can mean, that those data will never change again, the optimum for the outside user. But it might also mean, that only under certain circumstances those data will be changed. In this case it is important to know what "certain circumstances" means. And additionally it is important to know, how and where the used semantics are described. A resolvable pointer to them, of course is the solution which can be handled most easily.

All the steps within the life cycle of data can be stored as data provenance, containing at least information about the used objects, the produced objects and the applied action. There are two important use cases for data provenance: 1.) citation of data and all the actors involved in the production of the data. 2.) correct interpretation of the data.

The states changes of information objects as effects of actions are summarised in the following table, which can be included as provenance information. For example, a provenance tracking service may record information objects being processed, action types applied and resulting objects and some additional data and store that step.

Information Object	**Applied Action Types**	**Resulting Information Objects**
setup description of observation or measurement	specify measurement method	measurement description
persistent data (state: raw data)	data enrichment (multiple actions)	persistent data (diverse enriched states)
persistent data (state: mapped)	publish data	persistent data (state: published)
persistent data (state: finally reviewed)	mapping	persistent data (state: mapped)
persistent data (state: published)	process data	persistent data (state: processed)
persistent data (any states)	carry out backup	backup
persistent data (all enrichments)	final review	persistent data (state: finally reviewed)
measurement result	store data	persistent data (state: raw data)
measurement description	perform measurement or observation	measurement result
	specify sampling design	setup description of observation or measurement

SEVENTH FRAMEWORK PROGRAMME

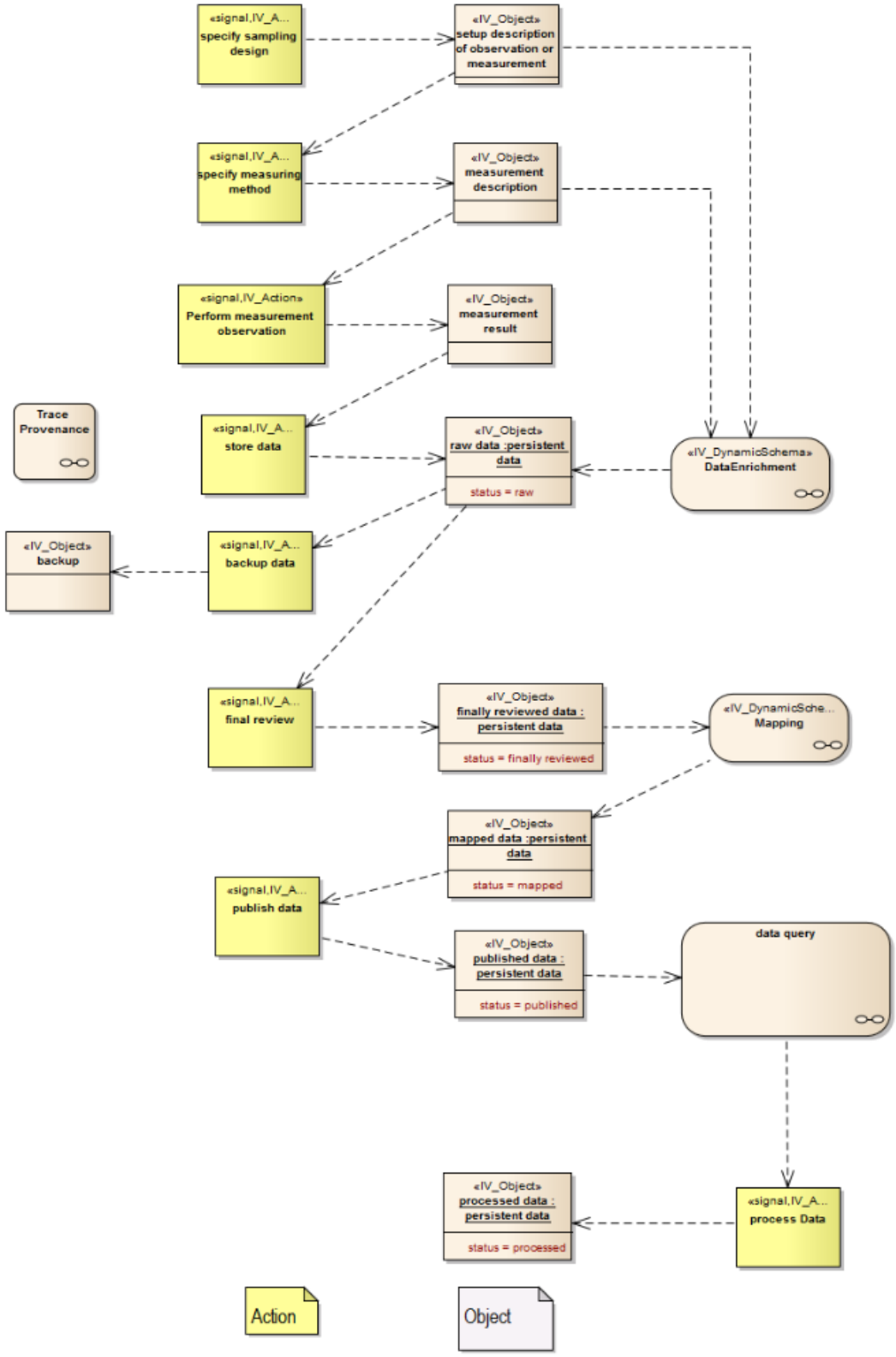


Figure 4.16: Dynamic Schemata Overview

3.2.2.2 Provenance Tracking

It is important to track state changes of information objects during their lifecyle. As illustrated in Figure 4.17, **track provenances** action is taken place at each point that action applied or any state changes of persistent data.

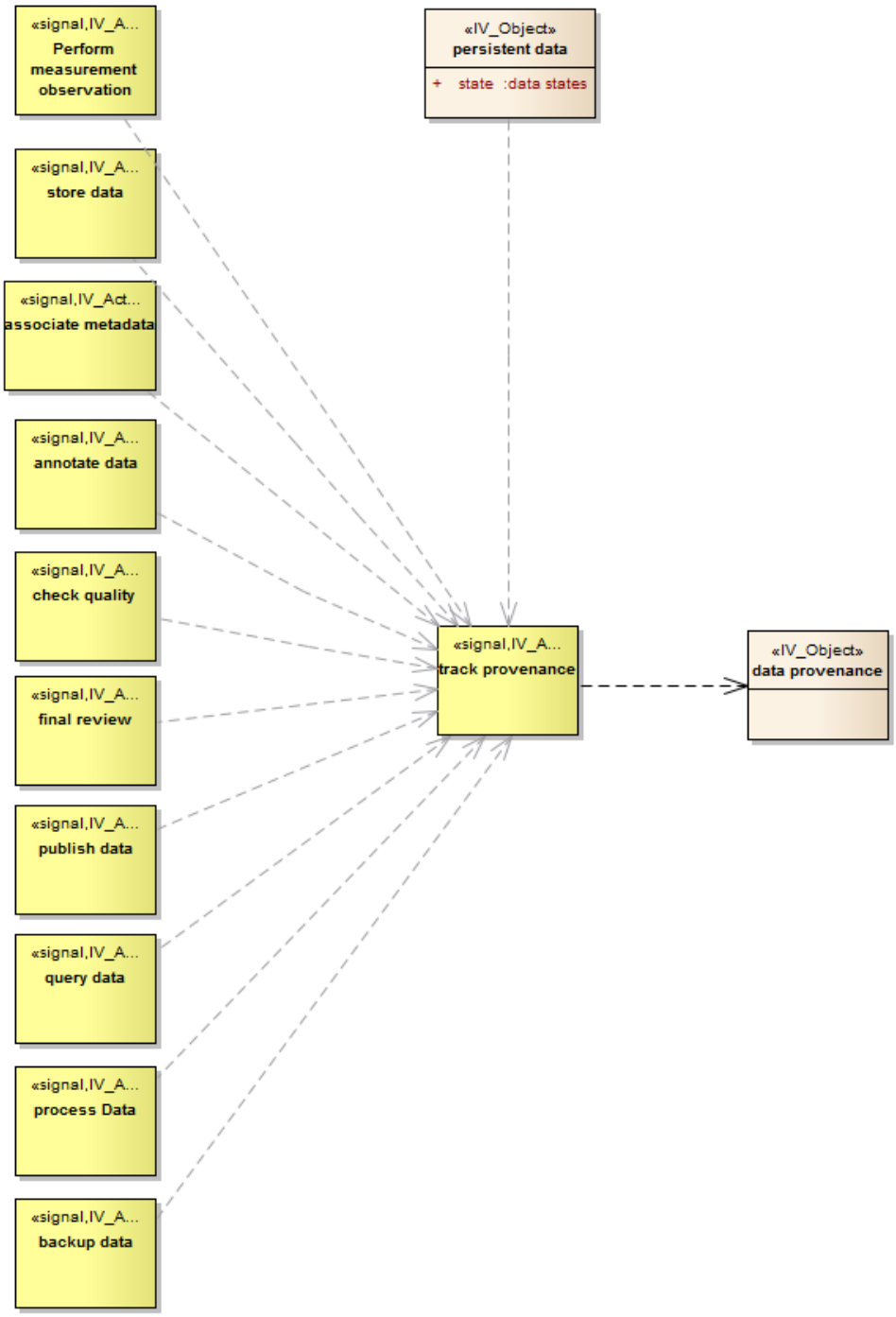


Figure 4.17: Tracing of Provenance

3.2.3 Static Schemata

Static Schemata specifies a minimum set of constraints when sharing data, in order to:

A. avoid loss of information around measurements and observations;
B. provide information about the meaning of data; and
C. make data and additional information available.

Three static schemata are specified, constraints for **data collection**, **data integration** and **data publication**.

3.2.3.1 Constraints for Data Collection

A collection of constraints applied to data collection is illustrated in Figure 4.18, which can help avoid information loss or wrong information to be drawn out.

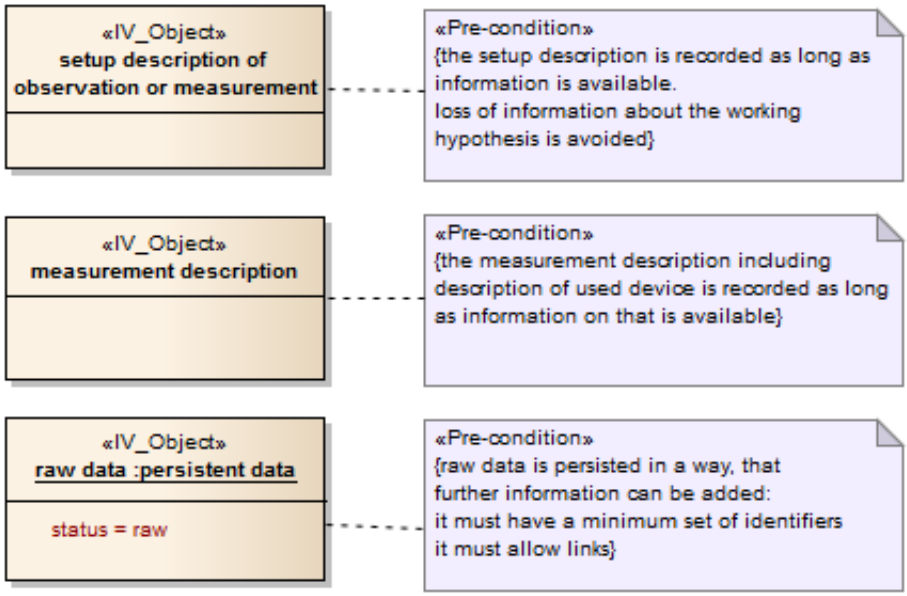


Figure 4.18: Constraints for Data Collection

3.2.3.2 Constraints for Data Integration

Figure 4.19 specifies the constraints for data integration, which is used for interpreting the meaning of data in order to help external data users correctly understand and map the semantics of data.

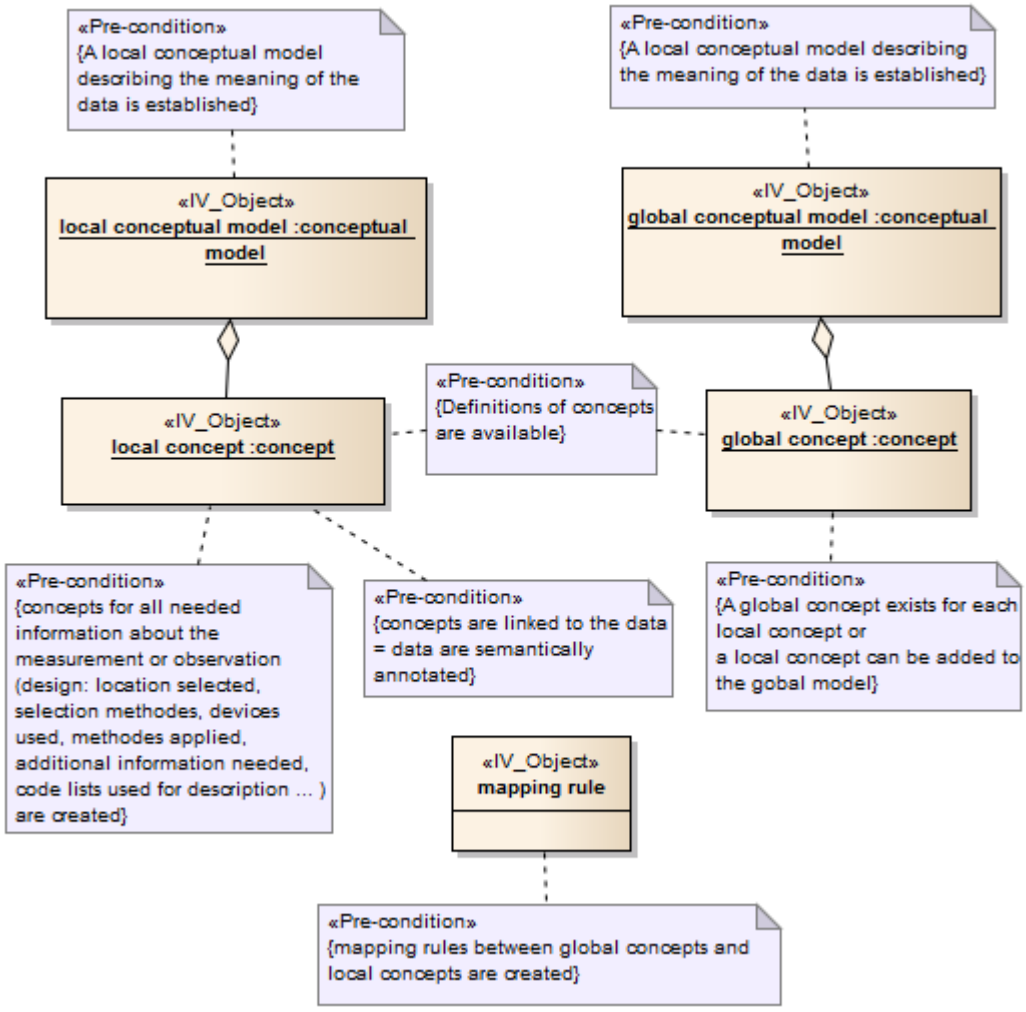


Figure 4.19: Constraints for Data Integration

3.2.3.3 Constraints for Data Publication

Constraints for data publication are described in Figure 4.20 which specifies pre-conditions and constraints necessary for preparing the data to be public accessed.

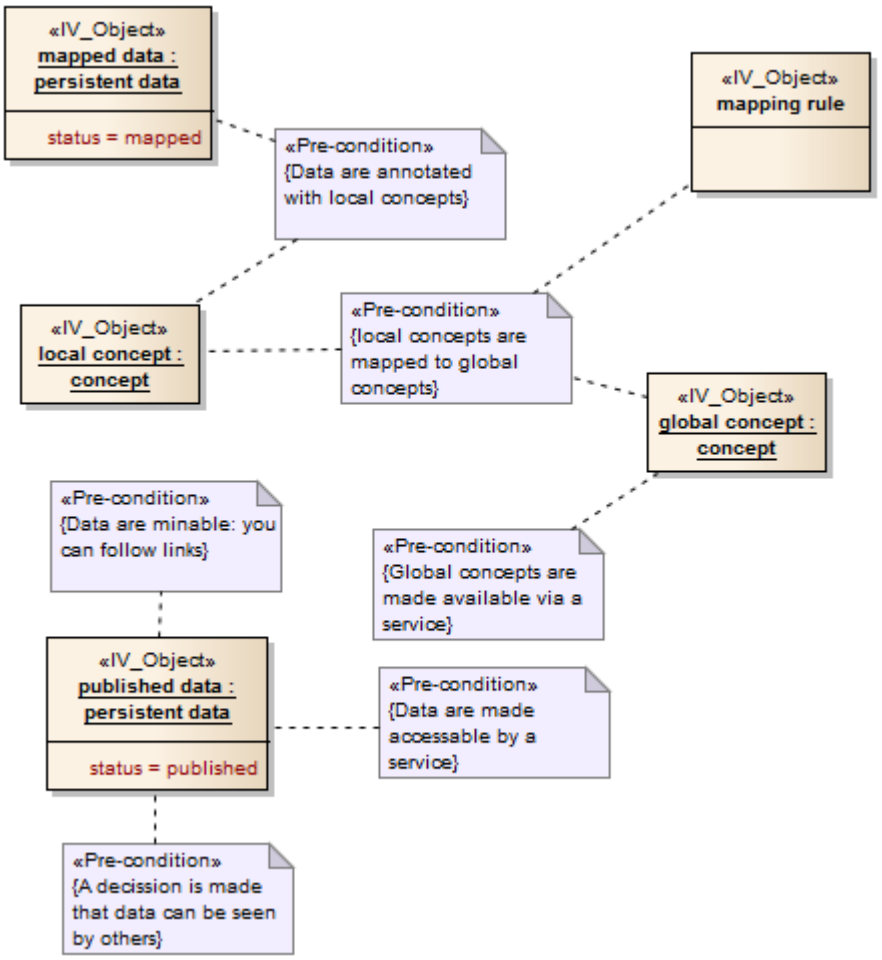


Figure 4.20: Constraints for Data Publication

3.2.4 Subsystems

For the purposes of easy observations, the defined model objects are regrouped into subsystems as defined in Section 2 Model Overview. Only 3 subsystems are discussed here:

- **Data Acquisition**, which consists of a group of information objects, and action types handled in the data acquisition subsystem;
- **Data Curation**, which consists of a group of information objects, and actions types handled in the data curation subsystem;
- **Data Access**, which consists of a group of information objects and actions types handled in the data access subsystems.

3.2.4.1 Data Acquisition

The information objects, and action types involved in the Data Acquisition subsystem are depicted in Figure 4.21, which supports the following functionalities:

- Measurement / Observation Design
- Observation
- Measurement

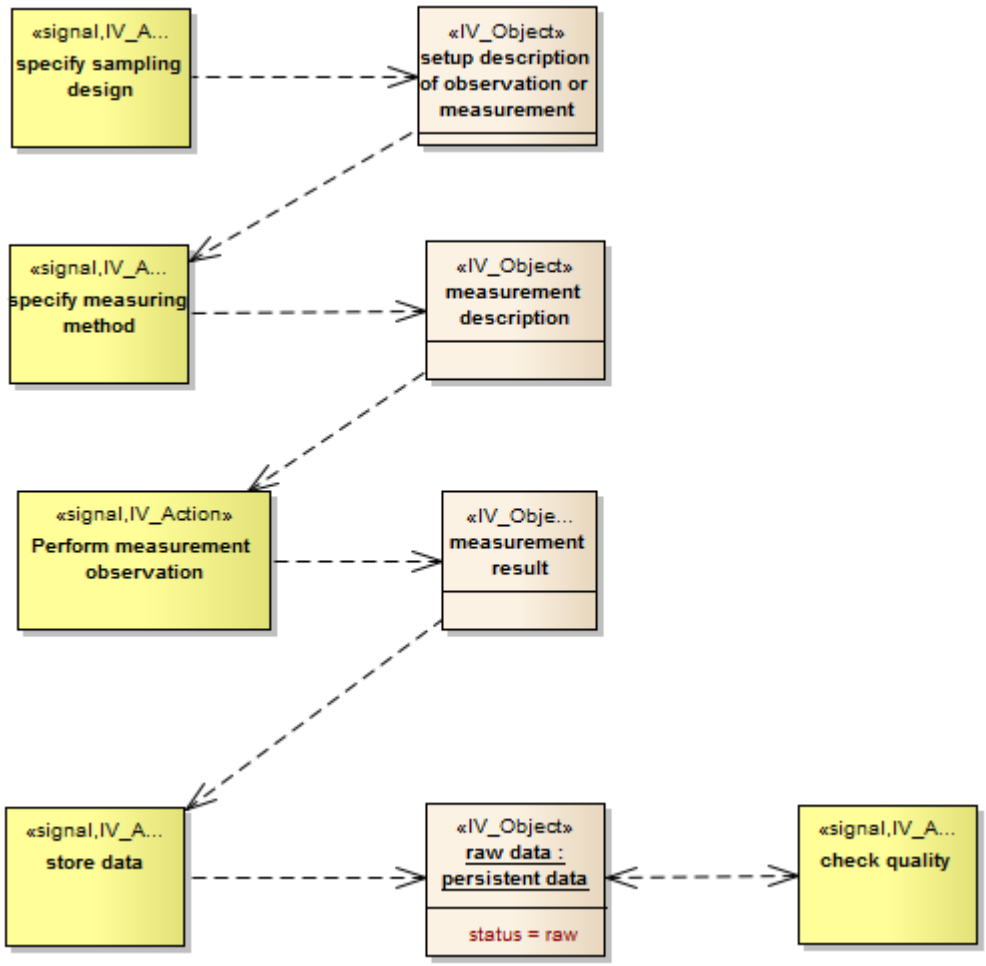


Figure 4.21: Information Specification of Data Acquisition Subsystem

A design for observation / measurements, is established and described in the setup description. Then the measurement method is specified including specifications for the measurement devices. The measurement or observation leads to results which are persisted (stored).

3.2.4.2 Data Curation

Figure 4.22 describes the information objects, and action types involved in the Data Curation subsystem, which support the following functionalities:

- persistent data preservation
- assignment of metadata
- semantics annotation
- quality assurance

- data identification - assignment of persistent identifier
- data backup

Raw data are collected by observations and measurements, and curated and stored in a data curation subsystem.

The following data curation operations are optional:

- QA annotation,
- obtain unique identifiers
- make a backup (mostly for long-term data persistence)
- connect the data to a (machine-readable) description - make semantic annotations - create contextual metadata
- Link the data to metadata, making data searchable

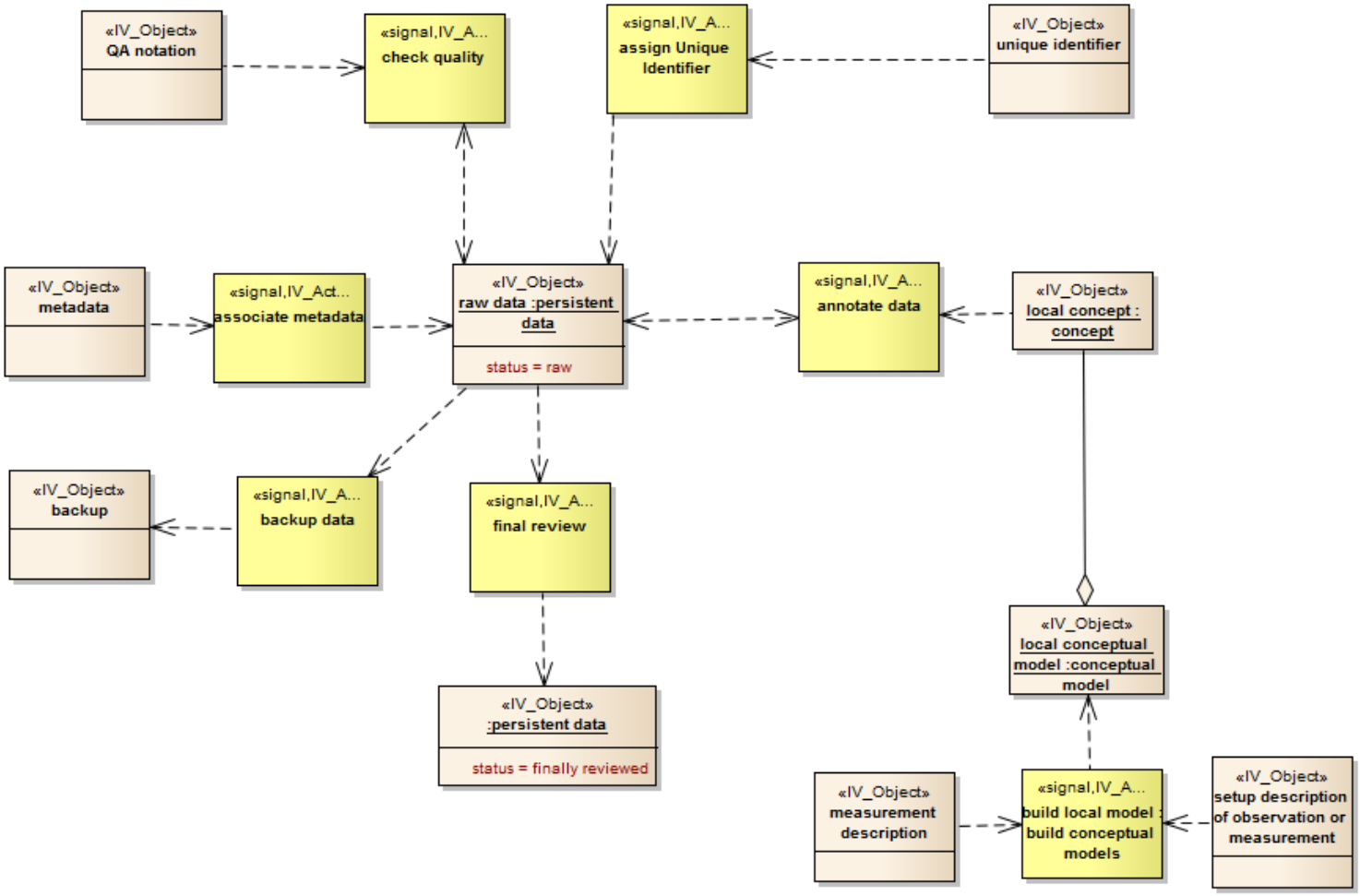


Figure 4.22: Information Specification of Data Curation Subsystem

3.2.4.3 Data Access

The information objects, and action types involved in the Data Access subsystem are shown in Figure 4.23. which supports the following functionalities:

- final review of data to be published
- data mapping (semantic mediation)
- data publication
- data processing

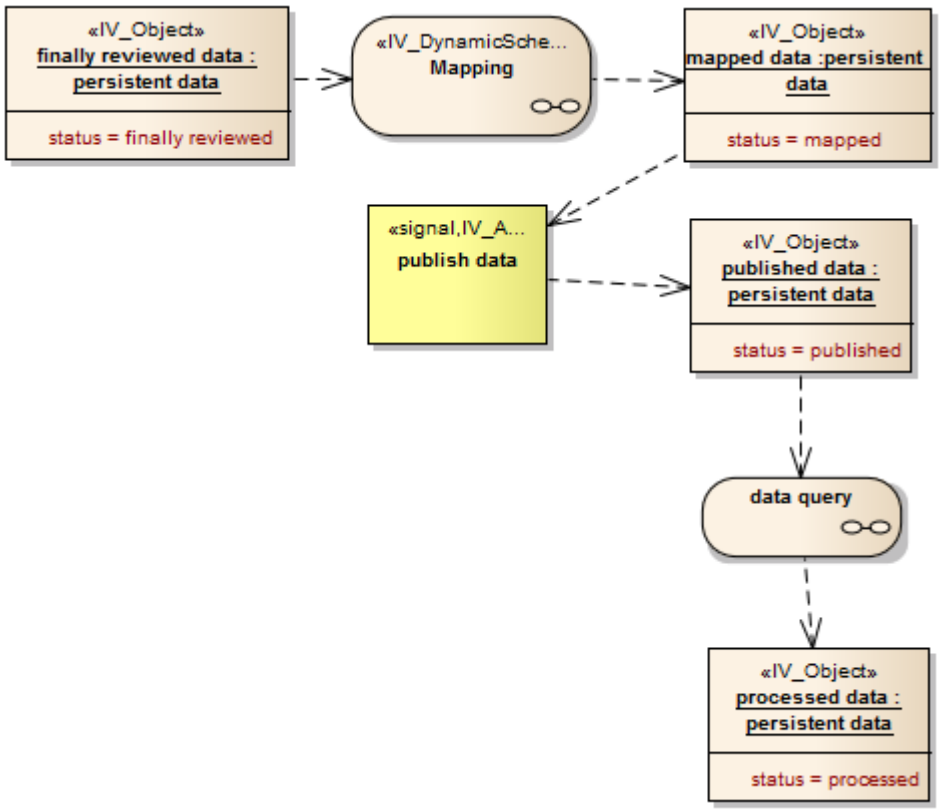


Figure 4.23: Information Specification of Data Access Subsystem

3.3 Computational Viewpoint

An environmental research infrastructure provides scientific data and the means to interact with that data. A number of different services and processes must be implemented if the infrastructure is to be able to acquire, store and provide access to a significant body of data. Indeed simply providing the ability to catalogue and annotate data, or to even visualise data requests, requires that a number of core functions be realised.

Each of the five subsystems of the ENVRI reference model must provide a number of computational objects which can be distributed across implemented services and resources in a concrete research infrastructure. Each object provides a number of interfaces by which its functionality can be accessed; each object also has facilities to invoke the functionality possessed by other objects by those

interfaces. The computational viewpoint of each of the five subsystems is provided along with descriptions of their computational objects and interfaces:

- **Data acquisition**
- **Data curation**
- **Data access**
- **Data processing**
- **Community support**

A number of reference interactions describing the interaction between objects in different subsystems is also provided:

- **Brokered data export**: the export of user-requested data;
- **Brokered data import**: the import of user-provided data;
- **Brokered data query**: the querying of curated data by users;
- **Citation**: the resolution of data and resources cited in publications;
- **Internal data staging:** the preparation of curated data for processing;
- **Processed data import**: the curation of results derived from processing;
- **Raw data collection**: the acquisition of raw data from integrated data sources;
- **Request verification**: verification of user data requests;
- **Resource registration:** the registration of new resources within the infrastructure;
- **Task verification:** verification of user processing requests.

The aggregation of these interactions forms a minimal computational model for the ENVRI reference model.

3.3.1 Computational modelling

The computational viewpoint of ODP is concerned with the modelling of computational objects and their interfaces. Each interface provides a set of invocations by which an object can be interacted with or by which an object can interact with another. Complex interactions of objects can be realised by the use of binding objects which encapsulate the coordination of multiple objects via multiple interfaces. Specific interactions can be specified in more detail, breaking them down into their constituent invocations.

Major computational interactions are modelled as collaborations between computational objects. Objects are linked via interfaces: **operation interfaces** are used to specify the service invocations which can be called between objects whilst **stream interfaces** describe data channels between objects by which bulk datasets can be moved.

- **Operation interfaces** are assumed to define all procedure calls (or service calls) necessary to permit collaboration between objects. In particular, it is assumed that procedures allow back-and-forth communication between objects allowing for (where applicable) complex negotiations and coordination actions.
- **Stream interfaces** are defined only where movement of significant volumes of data is desirable – 'significant' in this case being large enough to necessitate the used of dedicated file movement protocols. Data which can be wrapped within service invocations and responses are generally assumed to be transferred via operation interfaces. Note that 'stream' interfaces do not necessarily

imply that data is actually streamed; it should be assumed however that data channels created by stream interfaces can persist over long periods and out-live individual service invocations.

- The computational viewpoint of the ENVRI-RM accounts for the major computational objects expected within an environmental research infrastructure as well as the interfaces necessary for those objects to interact. Particular attention has been given to the interfaces between subsystems, where many critical functions can be expected to intercede in movements of data from one subsystem to another. All core reference points between subsystems focus on the delivery of data from one subsystem to another, and the services which must be invoked at some point during data transport. In each instance of data movement between subsystems, the computational objects involved can be divided into three groups concerning **control**, **delivery**, and **processing**, respectively.
- **Control** over data delivery is accorded to one or more oversight services, typically each representing a particular subsystem of the research infrastructure. Each control service is expected to prepare and configure the computational objects involved in the actual delivery of data. One service will act as the initiator for data movement, often on behalf of some external agency or internal scheduled workflow, and coordinate with the other control services. One service will invoke the binding object which represents the (abstract) interaction of processing services with the movement (and conversion) of datasets between subsystems.
- **Delivery** of data is achieved by establishing data channels between data sources and data destinations. A channel is described by a binding interaction involving a binding object which takes data streamed from a data source host object, invokes any necessary processing services required for data conversion or accounting, and sends the (converted) data on to the destination host object.
- **Processing** of data is performed at some point between data source and destination. A number of computational objects will be bound to the data delivery event representing services or resources which may be invoked by an infrastructure in order to verify, prepare or analyse data in transit.

Each reference point diagram (which can generally be mapped to points-of-reference between subsystems) involves a central binding object used to connect all computational objects which must interact at this reference point. The binding object serves to illustrate which computational objects (and thus computational functions) should be available at some point during the delivery of data; whether a particular object is invoked at the data source, destination or some other interim point is not considered to be important within the reference model.

3.3.2 Data Acquisition

The continuous acquisition of data requires the existence of an integrated network of instruments from which data can be extracted. Whether the instruments represent individual sensors, sensor arrays or something even further removed from the physical recording of measurements, an 'instrument' from the computational perspective represents a data source which produces new data over time and which can be directly queried for that data at any time. Integration, in this instance, refers to the fact that these instruments are considered part of the infrastructure (as opposed to being provided by an outside agency). Instruments can therefore be read from within the infrastructure without the need for the data access subsystem to verify the instrument's identity and validate any requests made.

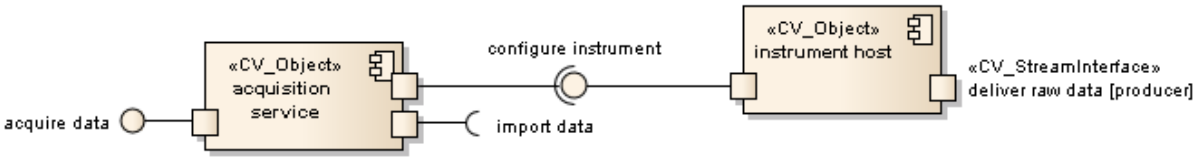


Figure 4.24: Computational Specification of Data Acquisition Subsystem

The data acquisition subsystem is computationally described as a set of **instrument hosts** (representing the computational functionality of instruments) monitored and managed by an **acquisition service**. The acquisition service fields all directives sent to the instrument network (via its **acquire data** interface), schedules data collection and performs any re-calibrations of instruments requested within the instruments' operation boundaries (via the **configure instrument** interface). It is also responsible for negotiating the import of data into the data curation subsystem via the **import data** interface. Under the oversight of the acquisition service, instrument hosts can **deliver raw data** to data resource hosts within the data curation subsystem in order to (continuously) move data directly from instruments to data stores.

acquisition service

Oversight service for an integrated data acquisition instrument network.

An acquisition service object collects the computational functions required to manage a network of data-collecting instruments integrated into an infrastructure. Any service or set of services implementing the acquisition service is responsible for integrating new instruments into the network, re-calibrating existing instruments (within the permitted operational bounds of those instruments) and managing the extraction of data from all instruments in the data acquisition network.

The acquisition service invokes two interfaces:

- **configure instrument** is used to configure or calibrate individual instruments.
- **import data** is used to negotiate the transfer of data into the data curation subsystem.

The acquisition service provides one interface:

- **acquire data** provides functions for controlling the data acquisition network: instrument calibration, instrument configuration, data acquisition requests and acquisition scheduling.

instrument host

An instrument (integrated raw data source) in the data acquisition subsystem.

An instrument is a source of raw data for a research infrastructure. Instruments are distinguished from other data sources in that they are fully integrated into a data acquisition network within an infrastructure's data acquisition subsystem, which allows their status to be continually monitored and their output directly channelled into a data curation system without the intercession of an intermediate broker. Instrument hosts collect the computational functions required to interact with instruments directly.

An instrument host can deliver raw data to the data curation subsystem (see subsection 3.3.13).

SEVENTH FRAMEWORK PROGRAMME

An instrument host provides one interface:

- **configure instrument** provides functions to adjust how an instrument collects data (if configurable) and to schedule data transfer from instrument to data resource.

3.3.3 Data Curation

Scientific data must be collected, catalogued and made accessible to all authorised users. The accessibility requirement in particular dictates that infrastructures provide facilities to ensure the easy availability of data products, generally by replication (avoiding data loss by system failure, as well as more optimised data retrieval for larger, distributed infrastructures), the use of persistent identifiers (providing a long-term means to precisely locate specific data-sets) and the use of metadata catalogues (providing a means to search for and discover data).

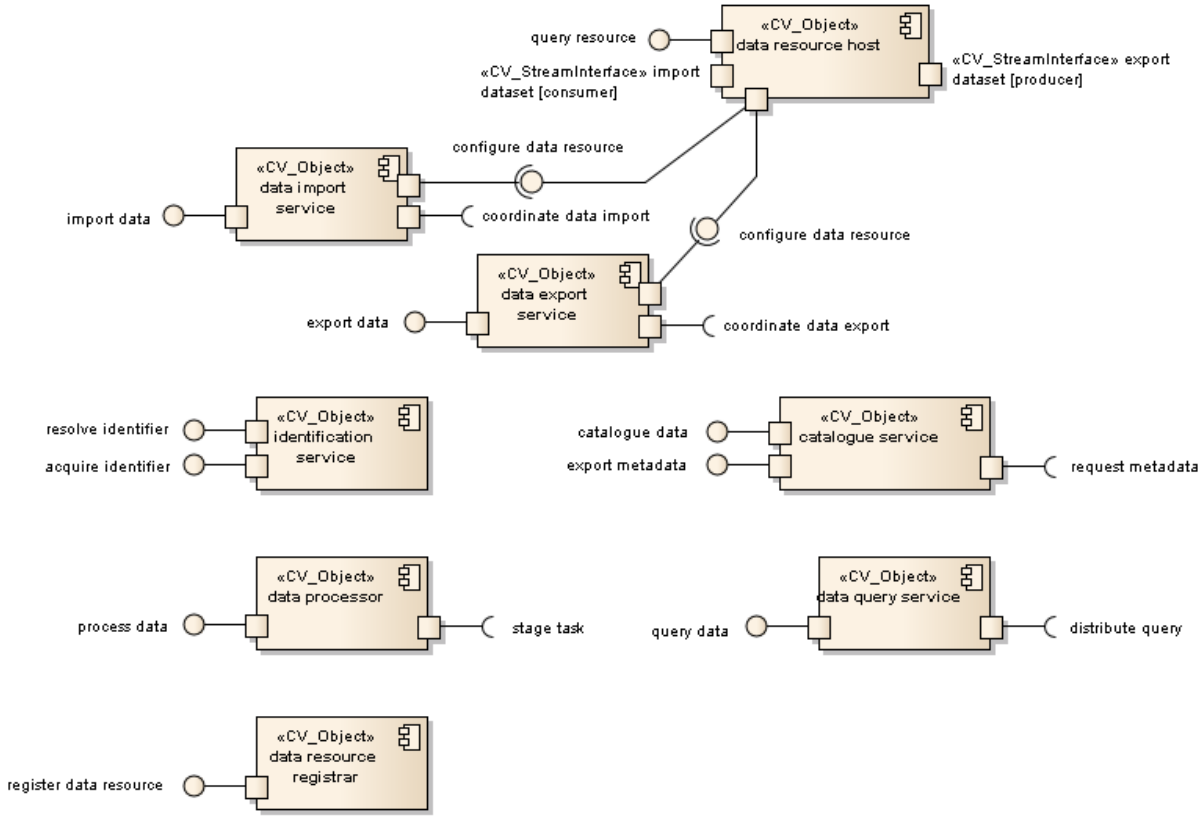


Figure 4.25: Computational Specification of Data Curation Subsystem

The data curation subsystem is computationally described as a set of **data resource hosts** (representing the computational functionality of data resources as presented to the larger infrastructure) monitored and managed by a number of curation services via their **configure data resource** interfaces. These curation services, which often interact directly with other subsystems like Data Acquisition and Data Access include:

- The **data import service**, concerned with managing of the inclusion of new data into the infrastructure. The import service negotiates the import of data from instruments and other data sources to selected data resources. The act of importing data requires that any unstructured data be

packaged into a format suitable for curated data, which includes the assignment of persistent identifiers and essential metadata. The import data service is invoked via its **import data** interface and coordinates all services involved in the import of new data via its **coordinate data import** interface; see the raw data collection (for raw data acquisition), brokered data import (for user-driven data imports) and processed data import (for curation of data processing results) reference interactions.

- The **data export service**, concerned with managing the retrieval of datasets from the infrastructure for use by external services and agents. The export service negotiates the export of data from curated resources to external locations provided that the data request is authorised and the agent making the request has been duly authenticated. Exporting data may require further re-packaging of data, aggregating data from different datasets and attaching all relevant metadata which might not otherwise be packaged with the data itself. The export data service is invoked via its **export data** interface and coordinates all services involved in the export of curated data via its **coordinate data export** interfaces; see the brokered data export (for user-driven data exports) and internal data staging (for staging data for data processing) reference interactions.
- The **data query service**, concerned with the querying of data resources for specific information, which may involve the aggregation of results from multiple datasets. Data requests must be authorised just as for exported data; some complex requests may require processing of retrieved data, invoking the services of the data processing subsystem. The data query service is invoked via its **query data** interface and coordinates all services involved in the querying of curated data via its **distribute query** interface; see the brokered data query reference interaction.

Data resource hosts represent logically distinct data stores with their own internal data management regimes. These stores can be queried directly (without the need to export whole datasets) via the **query resource** interface. All data resources must be able to **import datasets** and **export datasets** on demand.

catalogue service

Oversight service for cataloguing metadata with the data curation subsystem.

A catalogue service object collects the computational functions required to manage the construction and maintenance of catalogues of metadata associated with data-sets stored within the data curation subsystem of a research infrastructure. No assumption is made as to whether metadata is stored separately or integrated into data-sets. Exported data-sets should be packaged with their metadata, a responsibility assisted by the catalogue service.

The catalogue service may invoke one interface:

- **request metadata** is used to acquire information from other services which might be associated with a given dataset as metadata.

The catalogue service provides two interfaces:

- **catalogue data** provides functions for scheduling the harvest of metadata from data-sets in order to derive or update data catalogues.
- **export metadata** provides functions for obtaining all metadata associated with a data-set in order to package it for use outside of the research infrastructure.

SEVENTH FRAMEWORK PROGRAMME

data export service

Oversight service for export of data from the data curation subsystem.

A data export service object collects the computational functions required to coordinate the export of datasets from the data curation subsystem of a research infrastructure. In particular, it is responsible for locating requested data and selecting the data resources from which the data will be retrieved, then ensuring that the retrieved data is packaged with all pertinent metadata for use outside the infrastructure.

The data export service invokes two interfaces:

- **configure data resource** is used to schedule and configure data resources involved in data transfer.
- **coordinate data export** is used to coordinate all services involved in the export of data.

The data export service provides one interface:

- **export data** provides functions for requesting and negotiating the retrieval of datasets from the data curation subsystem. Invoked by an access broker on behalf of an external agent or service or internally for data staging; if the request has been authorised, a destination must be provided so that the export service can establish the delivery channel.

data import service

Oversight service for the import of new data into the data curation subsystem.

A data import service object collects the computational functions required to integrate new data into the data curation subsystem of a research infrastructure. The import service coordinates the collection of raw data delivered by the data acquisition subsystem; this entails the refactoring of data streams into discrete, identifiable datasets and the further delivery of those data-sets to suitable data resources for curation. The import service also handles the ingestion of user-provided datasets where permitted, and the curation of derived results generated by the data processing subsystem, provided that a well-defined interface for translation exists.

The data import service invokes two interfaces:

- **configure data resource** is used to schedule and configure data resources involved in data transfer.
- **coordinate data import** is used to coordinate all services involved in the import of data.

The data import service provides one interface:

- **import data** provides functions for requesting and negotiating the delivery of data into the data curation subsystem. Invoked by a number of different subsystems, the import service will establish a delivery channel for imported data from provided data sources. The provided data profile ensures that the format of data transferred is understood by all parties.

data processor

Used by the data curation subsystem to perform simple processing of newly imported / exported data and invoke the data processing subsystem for more complex tasks.

The data processor object encapsulates functions within the data curation subsystem to refactor and prepare data products going into and out of the subsystem. Importantly, the data processor also allows the data processing subsystem to interact with data integration and retrieval processes, permitting standard process workflows to be put in place between the data curation and other subsystems.

A data processor can invoke one interface:

- **stage task** is used to enlist the services of the data processing subsystem for more complex processing tasks.

A data processor provides one interface:

- **process data** provides functions for performing various inline processing tasks over data; this may be transformative (converting data from one format to another) or analytical (extracting characteristic metadata on data being moved into or out of the data curation subsystem).

data query service

Oversight service for the querying of data resources in the data curation subsystem.

The data query service collects the computational functions required to query data resources in the data curation subsystem of a research infrastructure. The query service is responsible for the distribution and aggregation of sub-queries where data from multiple datasets has been requested. Operations which actually require the movement of entire datasets onto external resources are handled by the data export service.

The data query service invokes one interface:

- **distribute query** is used to coordinate the interrogation of one or more data resources and compose the results.

The data query service provides one interface:

- **query data** provides functions for interrogating data resources and making query requests of datasets held within.

data resource host

A data store within the data curation subsystem.

A data resource object represents a data store integrated into the data curation subsystem of a research infrastructure. It contains the computational functionality required to store and maintain data products produced within the infrastructure, as well as to provide on demand access to authorised agents. A data resource must be able to interpret internal data requests, obey directives from curation services and adhere to data management policies such as (where applicable) versioning, replication and persistence.

A data resource host can **import** and **export datasets**.

A data resource host provides two interfaces:

- **configure data resource** is used to configure and schedule aspects of how a data resource interacts with the data curation subsystem at large; this includes connecting with other resources in order to import or export data.
- **query resource** is used to receive and respond to queries over curated data within a data resource.

data resource registrar

Registration service for integrating new data resources into the data curation subsystem.

The data resource registrar collects the computational functions required to register and configure new data resources within the data curation subsystem of a research infrastructure. The registrar ensures that new resources can be identified within the system and provide all expected functions of a data resource; this entails the deployment of a new data resource host object to handle all service invocations.

A data resource registrar provides one interface:

- **register data resource** provides functions for identifying and registering a resource as a data resource within the data curation subsystem of an infrastructure.

identification service

Oversight service for identifier assignment and resolution.

Persistent identifiers are generated by a service either provided within or outside of the infrastructure. The identification service object collects the functions required to interact with such a service, acquiring identifiers for all artefacts which require them and providing a resolution service for cited identities.

Different versions of artefacts, where maintained separately, are assumed to have different identifiers, but those identifiers should share a common root such that the family of versions of a given artefact can be retrieved in one transaction, or only the most recent (or otherwise dominant) version is returned.

An identification service provides two interfaces:

- **acquire identifier** provides an identifier for the given artefact.
- **resolve identifier** provides functions for identifying the artefact associated with a given identifier.

3.3.4 Data Access

A core responsibility of a research infrastructure is to provide access to the scientific data held within to the broader scientific community. This access can be provided in a number of ways: by allowing the export of data-sets, by allowing the data to be read via a client, by providing interactive visualisations via a portal, *etc*. Beyond the actual mechanism of data retrieval however is the issue of data discovery. Specific data-sets may be found via citation (the de-referencing of persistent identifiers), by browsing catalogues or by search over metadata. Data requests can be formulated which extract specific data elements from one or more datasets, returning the aggregation of all discovered results.

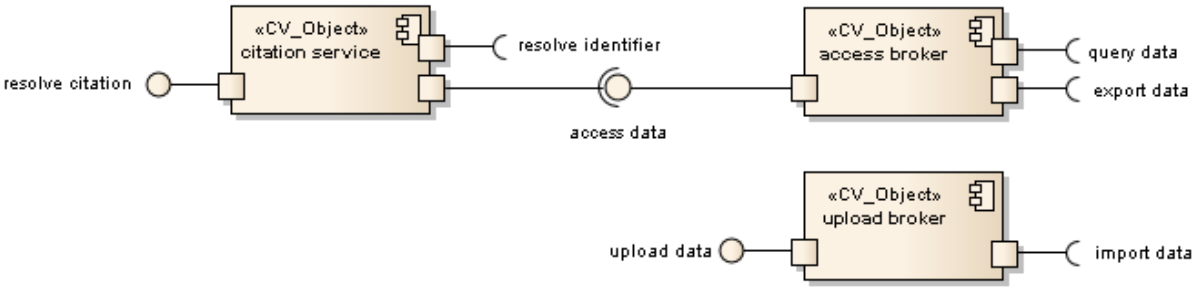


Figure 4.26: Computational Specification of Data Access Subsystem

The data access subsystem provides a number of data access brokers, which act as intermediaries for access to data held within the data curation subsystem. These brokers are responsible for verifying the agents making access requests and validating those requests before sending them on to the relevant data curation service.

- **Access brokers** are invoked via their **access data** interfaces; depending on the nature of the access request, they will either query the data curation subsystem via **query data** interfaces or request the export of data to an external resource via **export data** interfaces.
- **Upload brokers** are invoked via their **upload data** interfaces. If satisfied that the provided user data is suitable, then the upload broker will arrange the import of that data into the data curation subsystem via the **import data** interface.
- The **citation service** is invoked via its **resolve citation** interface, invoked whenever a citation to a persistent entity is de-referenced in a publication; the citation service will resolve any persistent identifier (via **resolve identifier**) and, if required, invoke the access broker in turn to retrieve information about the cited entity.

access broker

Broker for facilitating data access requests.

An access broker object intercedes between the data access subsystem and the data curation subsystem, collecting the computational functions required to make direct requests to data curation services on behalf of a user. It is the responsibility of an access broker to validate all requests and to verify the identity and access privileges of agents making requests. It is not permitted for an outside agency or service to access the data curation subsystem of a research infrastructure by any means other than via an access broker.

An access broker can invoke two interfaces:

- **export data** is used to request that a dataset be delivered to an external resource.
- **query data** is used to query data resources in the data curation subsystem.

An access broker provides one interface:

- **access data** provides functions for reading curated data; based on the nature of the request, responses can be generated within the infrastructure, or datasets can be produced for export.

citation service

Access service for resolving citations used in external publications to refer to artefacts within a research infrastructure.

A citation service provides one interface:

- **resolve citation** provides functions for identifying the entity being referred to within a published citation.

upload broker

Broker for facilitating data upload requests from external contributors.

An upload broker object intercedes between the data access subsystem and the data curation subsystem, collecting the computational functions required to upload user data into a research infrastructure for curation. It is the responsibility of an upload broker to validate all upload requests and to verify the identity and privileges of agents wanting to contribute data. It is not permitted for user data to be inserted into the data curation subsystem of a research infrastructure by any means other than via an upload broker.

An upload broker can invoke one interface:

- **import data** is used to request that a dataset be delivered to a data resource.

An upload broker provides one interface:

- **upload data** provides functions for uploading user data; based on the rights of the user, data will be packaged and brought into the data curation subsystem for storage and later access.

3.3.5 Data Processing

The processing of data can be tightly integrated into data handling systems, or can be delegated to a separate set of services invoked on demand; generally the more involved the processing, the more likely that separate resources will be required, particularly for tasks which require some form of high performance computing. The provision of dedicated processing services becomes significantly more important when large quantities (terabyte scale and higher) of data are being curated within a research infrastructure, especially for scientific data, which is often subject to extensive post-processing and analysis in order to extract new results. The data processing subsystem of an infrastructure encapsulates the dedicated processing services made available to that infrastructure, either within the infrastructure itself or delegated to a client infrastructure.

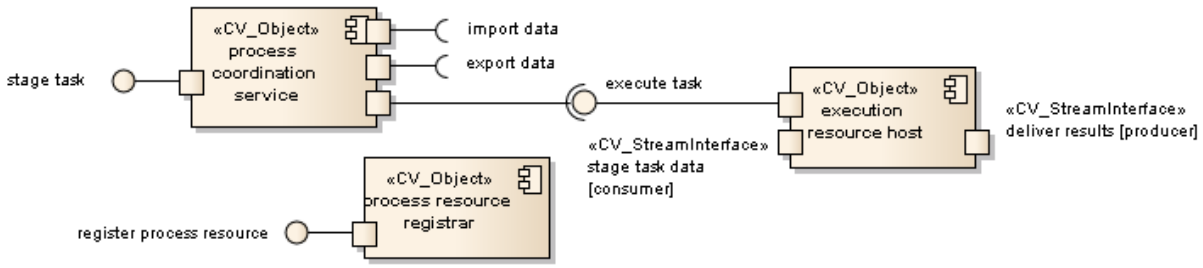


Figure 4.27: Computational Specification of Data Processing Subsystem

The data processing subsystem is computationally described as a set of **execution resource hosts** (representing the computational functionality of registered execution resources) monitored and managed by a **process coordination service**. The process coordination service delegates all processing tasks sent to the data processing subsystem (received via its **stage task** interface) to

particular execution resources, coordinates multi-stage workflows and initiates execution (via the **execute tasks** interfaces of selected execution resource hosts). Data may need to be staged onto individual execution resources and results retrieved for curation – this is handled by their hosts via the **stage task data** and **deliver results** stream interfaces respectively.

It is also possible to register new resources as execution resources, should the data processing subsystem need scaling up. This is handled by the **process resource registrar** via its **register process** resource interface.

execution resource host

Part of the execution platform provided by the data processing subsystem.

An execution resource host can be used to **stage task data** and **deliver results** back into the data curation subsystem.

An execution resource host provides one interface:

- **execute task** provides functions to deploy, schedule, configure and execute tasks on a given execution resource.

process coordination service

Oversight service for data processing tasks deployed on infrastructure execution resources.
A process coordination service can invoke three interfaces:

- **import data** is used to move derived data (such as process results) back into the data curation subsystem for long-term storage as a new (or modified) dataset.
- **export data** is used to request that datasets (or parts thereof) be staged onto execution resources for processing.
- **execute task** is used to coordinate the execution of data processing tasks on execution resources.

A process coordination service provides one interface:

- **stage task** provides functions for scheduling the execution of data processing tasks; this potentially includes complex workflows involving many (parallel) sub-tasks.

process resource registrar

Registration service for integrating new execution resources into the data processing subsystem.

A process resource registrar provides one interface:

- **register process resource** provides functions for registering a resource as a possible execution resource; this includes arranging the configuration of the new resource and the installation of any required programs.

3.3.6 Community Support

As important as the internal composition and management of an infrastructure is, equally important is how the broader scientific community interacts with that infrastructure. The principal way that prospective users interact with an infrastructure is via infrastructure-managed portals (generically referred to as 'scientific gateways' or ‘virtual research environments’). Given a multi-faceted infrastructure, there may exist numerous gateways into the infrastructure engineered towards

particular user roles and usecases. These gateways may provide additional user services beyond simple access to curated data and to data processing services; they may also provide 'social computing' services such as user profiling, reputation mechanisms and workflow sharing.

The community support subsystem also encompasses 'external' resources – resources not part of the infrastructure (or its federated client systems), but which are instead transitory resources temporarily brought in by a user as a data source or destination for bulk data transfers.

The community support subsystem plays host to a number of gateway host objects, representing the computational functionality of various assorted scientific gateways and portals.

- A **community gateway host** provides services to the broader scientific community. Via the gateway host, users can request data processing (via the **request task** interface), request access to data (via the **request access** interface, whether for data export or querying) and de-reference citations to persistent entities in publications (via the **resolve citation** interface).
- An **admin gateway host** provides services to infrastructure administrators and other 'privileged' users (such as the principal investigators associated with elements of the data acquisition network). Via the gateway host, users can register new data or execution resources (via the **register resource** interface).

The community support subsystem is considered to contain **external resources**, from which data can be imported into the infrastructure (via the **import data product** steam interface) or into which data can be exported from the infrastructure (via the **export data product** stream interface).

The community support subsystem also hosts the **authorisation service** (which can authorise requests via its **authorise request** interface) and the **authentication service** (which can authenticate users via its **authenticate user** interface); these services are invoked by *all* brokers before forwarding requests onto the data access subsystem.

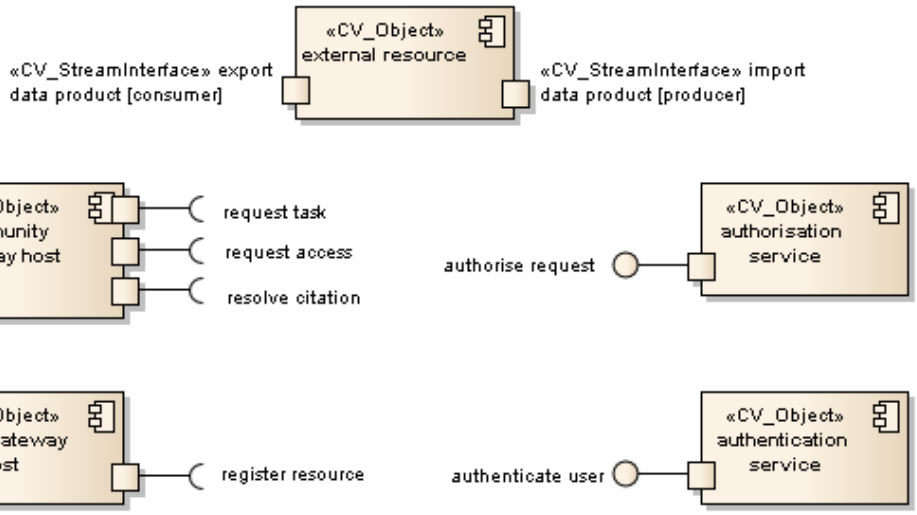


Figure 4.28: Computational Specification of Community Support Subsystem

admin gateway host

A gateway service or equivalent construct used to interact with the infrastructure in an administrative capacity.

An admin gateway host can invoke one interface:

- **register resource** is used to register new data or execution resources for use by the data curation or data processing subsystems respectively.

authentication service

Security service responsible for the authentication of external agents making requests of infrastructure services.

An authentication service provides one interface:

- **authenticate user** provides functions for identifying the source of a request and ascertaining their credentials.

authorisation service

Security service responsible for the authorisation of all requests made of infrastructure services by external agents.

An authorisation service provides one interface:

- **authorise request** provides functions for determining whether a given user is permitted to make the given requests, and whether the request is itself valid.

community gateway

A gateway service or equivalent construct used by the scientific community to interact with the infrastructure.

A community gateway host can invoke three interfaces:

- **request task** is used to request services from the data processing subsystem.
- **request access** is used to request access to data held within the data curation subsystem for querying or export.
- **resolve citation** is used to identify the artefacts referred to by citations embedded in publications outside of the infrastructure itself.

external resource

A resource not within the infrastructure acting temporarily as a data source or destination for data export.

An external resource is not assumed to provide or require any particular operational interfaces, but is assumed to be able to send and receive data via its **import data product** and **export data product** stream interfaces.

3.3.7 Brokered Data Export

Exporting data from curated data resources requires that the export be brokered by the data access subsystem before data can be retrieved from the data curation subsystem and delivered to a designated external resource. The interaction between the data access and data curation subsystems in this context is illustrated below.

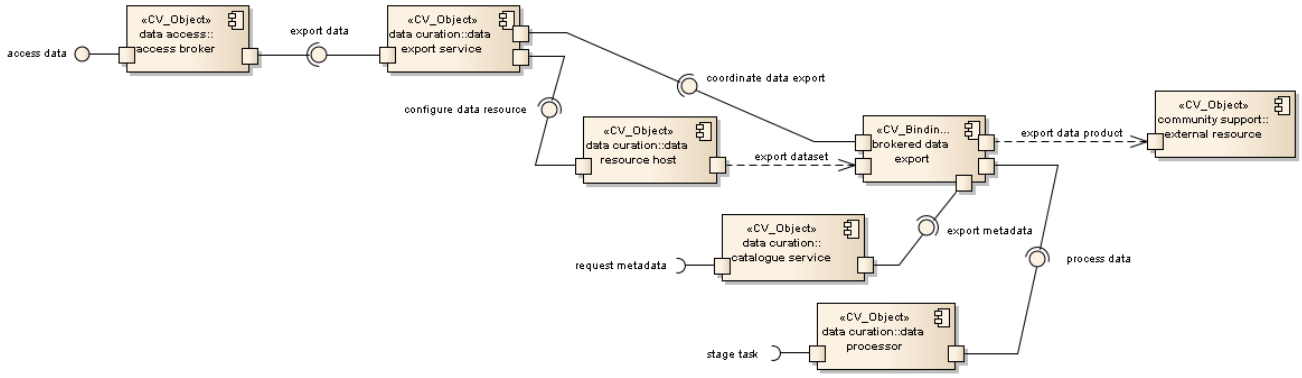


Figure 4.29: Brokered Data Export

Figure 4.29 provides three exposed interfaces: **access data**, by which the services of the data access broker are enlisted; **request metadata**, by which other parts of the infrastructure can contribute additional metadata information to package with the exported data product; and **stage task**, by which data processing services can be enlisted in the post-processing of exported data.

Any request for the export of datasets (base or derived) from a research infrastructure must be verified and validated by an **access broker** in accordance with the data access policies of the infrastructure. The retrieval of curated data is managed by collaboration between the access broker and the **data export service** via its **export data** interface. It is assumed that the access broker has been provided information about the data resource to which data is to be delivered; it is likewise assumed that this **external resource** is accessible via any and all required protocols. The data export service can prepare and configure data resources for data ingestion via the **configure data resource** interfaces on the respective **data resource hosts**.

The data export service oversees the interaction of services involved in the extraction of curated data out of the data curation subsystem and the establishment of delivery channels between external resources and internal curation resources – this is represented by the **coordinate data export** interface to the **brokered data export** binding object. The role of the brokered data export binding object is to link all computational objects involved in the retrieval of a data product from the data curation subsystem and the extraction of all pertinent metadata.

Given a delivery channel established between an external resource and a data resource host, overseen by the data export service, all curated data must be re-packaged into a self-describing data product which can be examined independently of the research infrastructure. Any externally-meaningful metadata associated with exported data must be packaged with the extracted data product. In certain circumstances, there may also be additional post-processing required in order to ensure that the data fulfils the original access request and does not violate any additional security constraints. The extraction of catalogued metadata is handled by the **catalogue service** invoked via the **export metadata** interface. Any post-processing is handled by the **data processor** invoked via the **process data** interface; the data processor may use its **stage task** interface to further invoke services provided by the data processing subsystem if necessary to carry out its function.

3.3.8 Brokered Data Import

Importing data from sources other than the acquisition network requires that the import be brokered by the data access subsystem before data can be delivered into the data curation subsystem. The interactions between the two subsystems in this context is illustrated below.

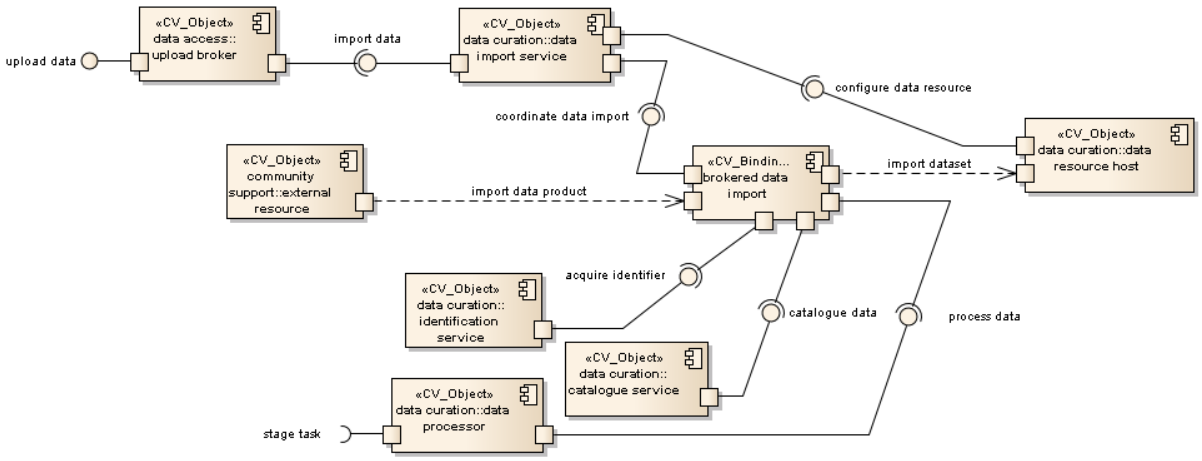


Figure 4.30: Brokered Data Import

Figure 4.30 provides two exposed interfaces: **upload data**, by which the services of the data upload broker are enlisted; and **stage task**, by which data processing services can be enlisted in the pre-processing of newly curated data.

Any request to upload data into a research infrastructure must be verified and validated by an **upload broker** in accordance with the data access policies of the infrastructure. The curation of imported data is managed by collaboration between the upload broker and the **data import service** via its **import data** interface. It is assumed that the upload broker has been provided information about the data source from which data is to be extracted; it is likewise assumed that this **external resource** is accessible via any and all required protocols. The data import service can prepare and configure data resources for data ingestion via the **configure data resource** interfaces on the respective **data resource hosts**.

The data import service oversees the interaction of services involved in the ingestion of imported data into the data curation subsystem and the establishment of delivery channels between external resources and internal curation resources – this is represented by the **coordinate data import** interface to the **brokered data import** binding object. The role of the brokered data import binding object is to link all computational objects involved in the delivery of external data products into the data curation subsystem and its conversion (where necessary) into curated datasets stored within data resources.

Given a delivery channel established between an external resource and a data resource host, overseen by the data import service, any data product must be re-packaged into an infrastructure-compliant dataset with its own persistent identifier. Moreover any metadata stored within the data product or evident in its provenance should be extracted and added to all pertinent catalogues. In certain circumstances, there may also be additional pre-processing required before the data can be stored. The assignment of persistent identifiers is handled by the **identification service** invoked via its **acquire**

identifier interface. The cataloguing of extracted metadata is handled by the **catalogue service** invoked via its **catalogue data** interface. Any pre-processing is handled by the **data processor** invoked via its **process data** interface; the data processor may use its **process data** interface to further invoke services provided by the data processing subsystem if necessary to carry out its function.

3.3.9 Brokered Data Query

Querying curated data resources requires that the request be brokered by the data access subsystem before any results will be retrieved from the data curation subsystem and delivered to the client from which the source came. The interaction between the data access and data curation subsystems in this context is illustrated below.

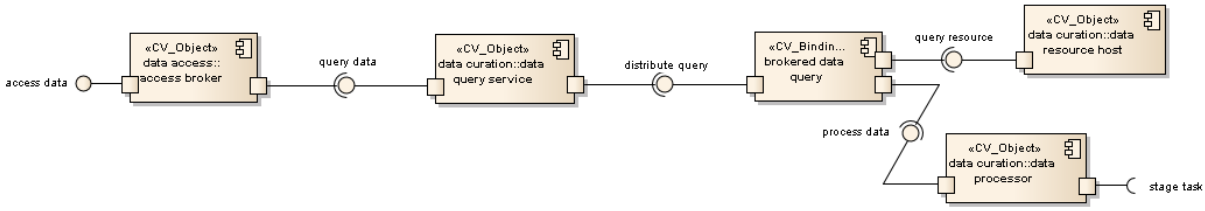


Figure 4.31: Brokered Data Query

Figure 4.31 provides two exposed interfaces: **access data**, by which the services of the data access broker are enlisted; and **stage task**, by which data processing services can be enlisted in the processing of the query should any such extraordinary processing be required.

Any data request made to a research infrastructure must be verified and validated by an **access broker** in accordance with the data access policies of the infrastructure. The discovery of curated data is managed by collaboration between the access broker and the **data query service** via its **query data** interface. Any results not requiring a full export of a (derived) dataset are returned via this interface. The data query service is then responsible for distributing the query across all necessary data resources (which may be greater than one for complex data requests) via the **distribute query** interface to the **brokered data query** binding object. The role of the brokered data query binding object is to link all computational objects involved in the aggregation of results to a data query request including possibly objects belonging to the data processing subsystem in complex cases.

Individual data resources identified as having required data by the data query service are queried via the **query resource** interfaces of their respective **data resource host** objects. If processing of data (either before or after aggregation of results) is required, the **data processor** can be invoked via its **process data** interface; for particularly complex requests, the data processor can invoke the services of the data process subsystem at large via the **stage task** interface.

3.3.10 Citation

The citation of datasets involves reference to persistent identifiers assigned to objects by a research infrastructure. Such citations are resolved by referring back to the infrastructure, which can then return a report describing the data cited. This simple interaction is illustrated below.

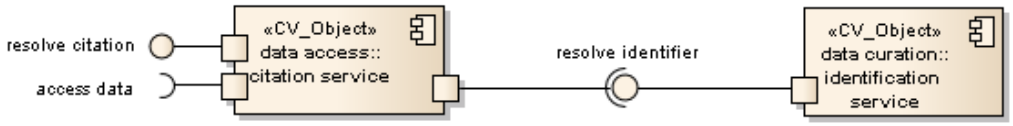


Figure 4.32: Citation

Figure 4.32 provides two exposed interfaces: **resolve citation**, by which the link between persistent identifiers and data objects is established; and **access data**, by which datasets referred to by a citation can be examined or retrieved.

The citation of datasets is managed by the **citation service** in the data access subsystem, which extracts the persistent identifier embedded in a given citation and then refers to the **identification service** in the data curation subsystem via its **resolve identifier** interface.

3.3.11 Internal Data Staging

The internal staging of data within an infrastructure for processing requires coordination between the data processing subsystem (which handles the actual processing workflow) and the data curation subsystem (which holds all scientific datasets within the infrastructure). The interactions between the two subsystems in this context is illustrated below.

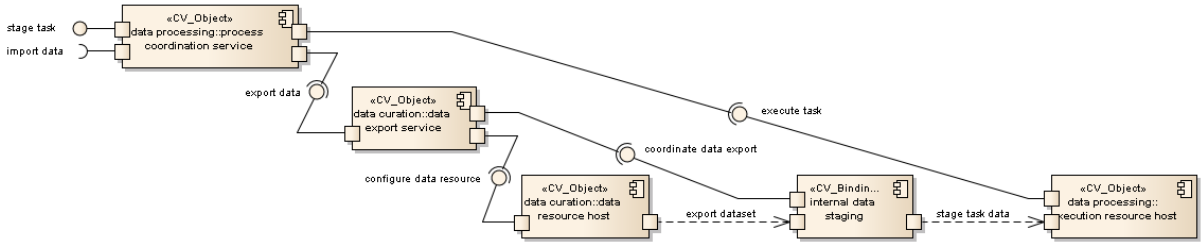


Figure 4.33: Internal Data Staging

Figure 4.33 provides two exposed interfaces: **stage task**, by which a processing workflow is specified and scheduled; and **import data**, by which derived datasets and results produced in processing can be formally ingested into the infrastructure.

Internal data staging is directed by the **process coordination service** in accordance with the task specifications given to it and any available infrastructure resources. The export of curated datasets onto execution resources is managed by collaboration between the process coordination service and the **data export service** via its **export data** interface. The process coordination service can configure execution resources and initiate processing of selected execution resources within the infrastructure by invoking the **execute task** interfaces of the **execution resource hosts** representing those resources. The data export service can prepare data resources for data export via the **configure data resource** interfaces on their respective **data resource hosts**.

The data export service oversees the interaction of services involved in the delivery of curated data into the data processing subsystem and the establishment of delivery channels between data and

execution resources – this is represented by the **coordinate data export** interface to the **internal data staging** binding object. The role of the internal data staging binding object is to link all computational objects involved in the staging of curated data onto execution resources. Note that it is perfectly valid for a data resource to also be an execution resource, likely for more efficient conduct of standard analyses on datasets. In this case, the 'staging' of data becomes trivial, with the data export service merely confirming that the data is suitably prepared for the processing task to be instigated by the process coordination service.

Once data has been confirmed as having been staged onto a suitable execution resource, data processing can proceed on that resource and results imported back into the data curation subsystem either as revisions of existing datasets or as new derived data.

3.3.12 Processed Data Import

The formal ingestion of data derived during internal processing of existing datasets within an infrastructure requires coordination between the data processing subsystem (which produces the new results) and the data curation subsystem (which is responsible for all scientific datasets within the infrastructure). The interactions between the two subsystems in this context is illustrated below.

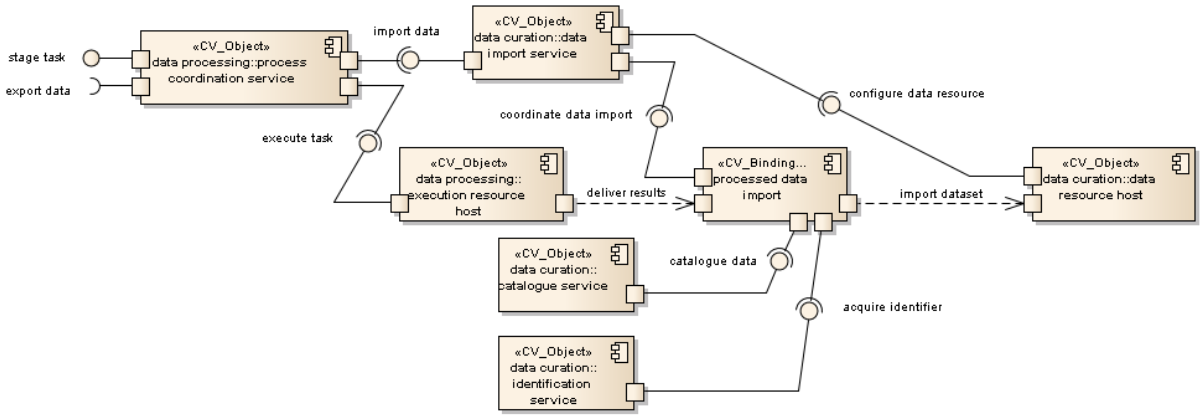


Figure 4.34: Processed Data Import

Figure 4.34 provides two exposed interfaces: **stage task**, by which a processing workflow is specified and scheduled; and **export data**, by which more data can be extracted from the data curation subsystem for further processing.

Processed data import is directed by the **process coordination service** in accordance with the task specifications given to it and the existence of uncurated results on any of the infrastructure resources it manages. The retrieval of derived datasets from execution resources is managed by collaboration between the process coordination service and the **data import service** via its **import data** interface. The process coordination service can configure execution resources and initiate extraction of derived data from selected execution resources within the infrastructure by invoking the **execute task** interfaces of the **execution resource hosts** representing those resources. The data import service can prepare selected data resources for data ingestion via the **configure data resource** interfaces on their respective **data resource hosts**.

The data export service oversees the interaction of services involved in the delivery of derived data into the data curation subsystem and the establishment of delivery channels between data and execution resources – this is represented by the **coordinate data import** interface to the **processed data import** binding object. The role of the processed data import binding object is to link all computational objects involved in the retrieval of processed data from execution resources. Note that it is perfectly valid for an execution resource to also be the data resource which is to curate a given set of results. In this case, the 'retrieval' of data becomes trivial, with the data import service merely confirming that the data is suitably prepared for curation.

Given a delivery channel established between an execution resource host and a data resource host, overseen by the data import service, any derived data product must be re-packaged into an infrastructure-compliant dataset with its own persistent identifier. Moreover any metadata stored within the data product or evident in its provenance should be extracted and added to all pertinent catalogues. The assignment of persistent identifiers is handled by the **identification service** invoked via its **acquire identifier** interface. The cataloguing of extracted metadata is handled by the **catalogue service** invoked via its **catalogue data** interface. It is assumed that any further processing of data required for curation is handled as part of the data processing activity already extant when producing the data.

3.3.13 Raw Data Collection

The collection of raw scientific data requires coordination between the data acquisition subsystem (which extracts the raw data from instruments) and the data curation subsystem (which packages and stores the data). The interactions between the two subsystems in this context is illustrated in Figure 4.36.

Figure 4.36 provides two exposed interfaces: **acquire data**, by which the behaviour of the acquisition service can be configured; and **stage task**, by which data processing services can be enlisted in the pre-processing of newly curated data.

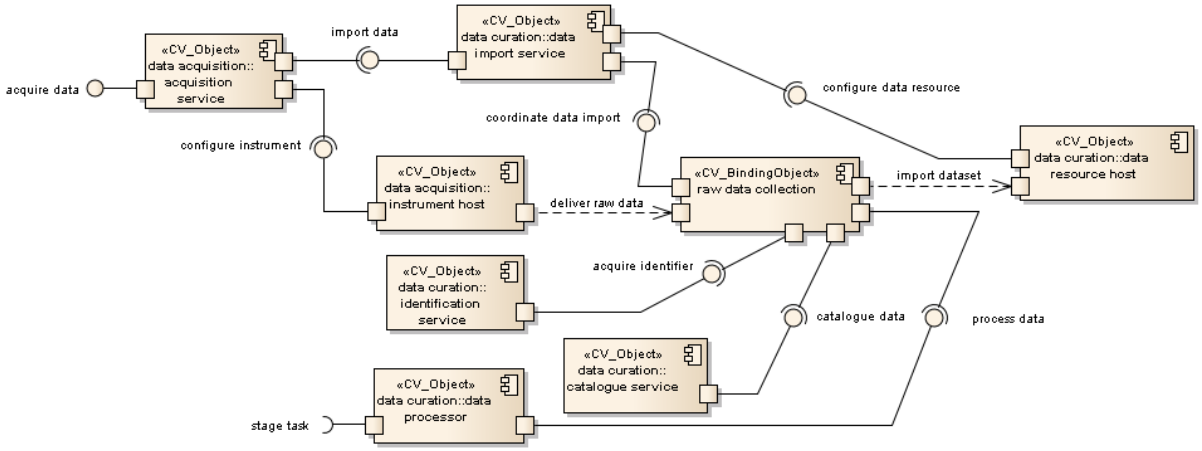


Figure 4.35: Raw Data Collection

Raw data acquisition is motivated by the **acquisition service** in accordance with its configuration. The curation of acquired data is managed by collaboration between the acquisition service and the **data import service** via its **import data** interface. The acquisition service can configure and calibrate the

SEVENTH FRAMEWORK PROGRAMME

behaviour of instruments in the acquisition network by invoking the **configure instrument** interface for the **instrument hosts** representing those instruments. The data import service can prepare and configure data resources for data ingestion via the **configure data resource** interface on the respective **data resource hosts**.

The data import service oversees the interaction of services involved in the ingestion of raw data into the data curation subsystem and the establishment of delivery channels between instruments and data resources – this is represented by the **coordinate data import** interface to the **raw data collection** binding object. The role of the raw data collection binding object is to link all computational objects involved in the delivery of raw data into the data curation subsystem and its conversion into curated datasets stored within data resources.

Given a delivery channel established between an instrument host and a data resource host, overseen by the data import service, raw data must be packaged into datasets in accordance with data curation policy. Each dataset should be assigned a persistent identifier and appropriate metadata describing its nature and provenance and subject to any pre-processing dictated by the research infrastructure. The assignment of persistent identifiers is handled by the **identification service** invoked via its **acquire identifier** interface. Any metadata generated should be catalogued – this is handled by the **catalogue service** via its **catalogue data** interface. Any pre-processing is handled by the **data processor** invoked via its **process data** interface; the data processor can use its **stage task** interface to further invoke services provided by the data processing subsystem if necessary to carry out its function.

3.3.14 Request Verification

Making a request for data movement to a research infrastructure requires that the request be brokered by the data access subsystem, which entails that the source of the request be authenticated and the request itself authorised. The interaction between a community gateway or portal and the data access subsystem in the context of data management is illustrated below.

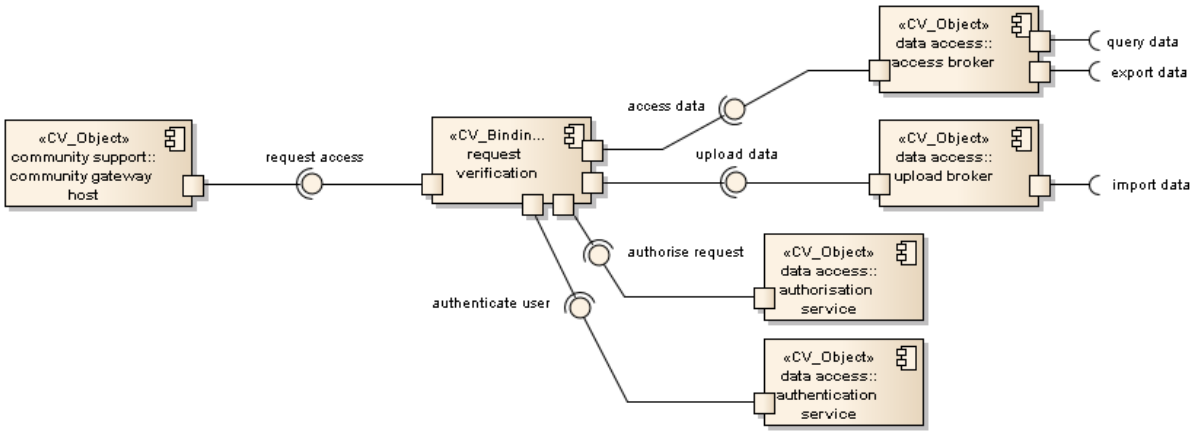


Figure 4.36: Request Verification

Figure 4.36 provides three interfaces into a research infrastructure depending on the data request being made of it: **query data**, by which the scientific data held by the infrastructure can be queried; **export data**, by which curated datasets can be retrieved from the infrastructure and delivered to a particular

SEVENTH FRAMEWORK PROGRAMME

location; and **import data**, by which data held by external agents or services can be brought into the infrastructure for curation.

Requests are assumed to be made via a protocol exposed by some sort of community tool such as a scientific gateway, represented by a **community gateway host**. Data requests are made using the **request access** interface of a **request verification** binding object. The role of a request verification binding object is to link together all computational objects involved in verifying and then passing on a data request to the data access subsystem. Thus the intercession of the **authentication service** (via its **authenticate user** interface) and the **authorisation service** (via its **authorise request** interface) is required. Only once these services have verified the request as being permissible can the request be passed on to the appropriate broker.

Requests involving the reading or export of datasets held within the data curation subsystem are brokered by an **access broker** via its **access data** interface. Requests involving the uploading of data into the data curation subsystem are brokered by an **upload broker** via its **upload data** interface. Complex requests may involve multiple brokers acting in parallel.

3.3.15 Resource Registration

The registration of new resources (whether to support data curation or data processing) requires that a request be made via suitable administrative channels, authorised and handled by some kind of registration service situated in the affected subsystem. The interaction between an administrative gateway or portal and the relevant subsystems in this context is illustrated below.

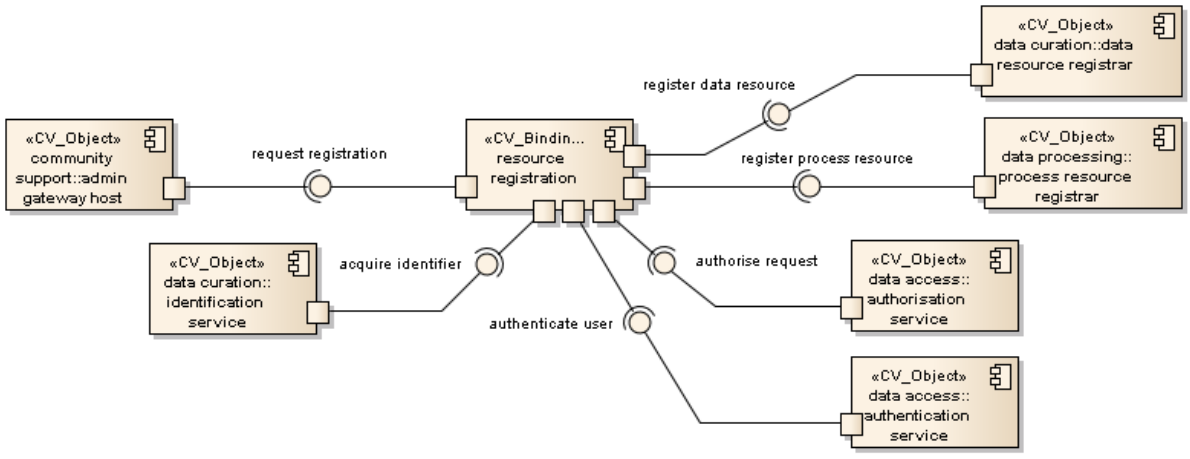


Figure 4.37: Resource Registration

Resource registration is considered to be a self-contained interaction. An administrative gateway represented by an **admin gateway host** invokes the **request registration** interface of a **resource registration** binding object. The role of the resource registration binding object is to link all computational objects involved in the registration of resources. Registration requires that the source of the registration request be authenticated, the request itself be authorised and the resource being registered be assigned a persistent identifier within the infrastructure and where appropriate an identifier which can identify the resource outside of it. Authentication is handled by the **authentication service** via its **authenticate user** interface. Authorisation is handled by

SEVENTH FRAMEWORK PROGRAMME

the **authorisation service** via its **authorise request** interface. Identification is handled by the **identification service** via its **acquire identifier** interface.

If a data resource is being registered, then the resource profile is passed on to the data curation subsystem; specifically a **data resource registrar** via its **register data resource** interface. If a processing or execution resource is being registered, then the resource profile is passed on to the data processing subsystem; specifically a **process resource registrar** via its **register process resource** interface. It is possible for a resource to be both a data resource and an execution resource, in which case both subsystems must be informed and must separately ensure adequate configuration.

3.3.16 Task Verification

Making a request for data processing to a research infrastructure requires that the request be brokered by the data access subsystem, which entails that the source of the request be authenticated and the request itself authorised. The interaction between a community gateway or portal and the data access subsystem in the context of data processing is illustrated below.

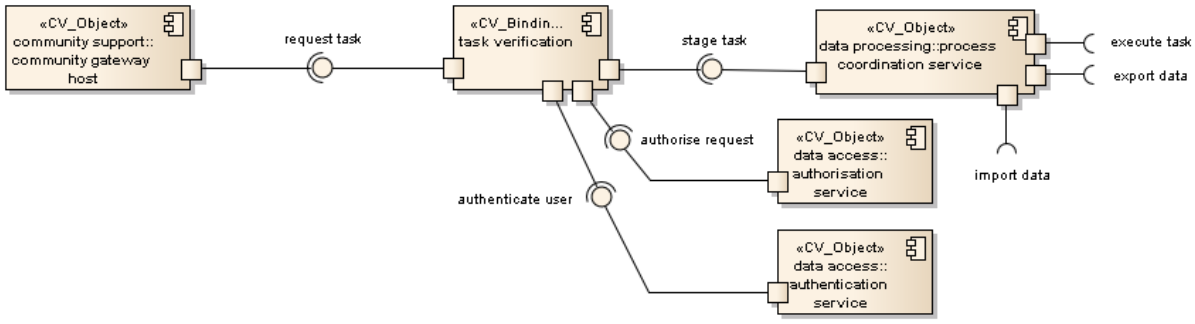


Figure 4.38: Task Verification

Figure 4.38 provides three interfaces into a research infrastructure depending on the requirements of the request being made of it: **execute task**, by which requested tasks are scheduled to be executed on resources belonging to the infrastructure; **export data**, by which curated datasets required for task execution are moved onto execution resources; and **import data**, by which results generated by data processing are moved into the data curation subsystem.

Requests are made via a scientific 'gateway' (or other tool presented to the community), represented by a **community gateway host**. Data processing requests are made using the **request task** interface of a **task verification** binding object. The role of a task verification binding object is to link together all computational objects involved in verifying and then passing on a task request to the data processing subsystem. Thus the intercession of the **authentication service** (via its **authenticate user** interface) and the **authorisation service** (via its **authorise request** interface) is required. Only once these services have verified the request as being permissible can the request be passed on to the **process coordination service** via its **stage task** interface.

All data processing tasks are overseen by the process coordination service, which coordinates the execution of individual tasks via its **execute task** interface, stages data via its **export data** interface and arranges the curation of any results produced via its **import data** interface. A complex task may involve multiple invocations of these interfaces.

4 CONCLUSION AND FUTURE WORK

The ENVRI Reference Model is a work in progress. Currently, attention is focused on three of the five ODP viewpoints: enterprise, information and computational. The remaining viewpoints of engineering and technology have been deferred to a later date.

Much work remains. Stronger correspondence between the three primary viewpoints is necessary to ensure that the three sub-models are synchronised in concept and execution. Further refactoring of individual components and further development of individual elements is to be expected as well. Further development of the presentation of the model is also essential, in order to both improve clarity to readers not expert in ODP and in order to promote a coherent position. In the immediate next step, the following tasks are planned:

Validation

The reference model will be validated from several aspects.

1. Usability. The users from different RIs will be invited to use the reference model to describe the research infrastructures in the ENVRI. The feedback will be collected and analysed to improve the definition of the reference model.
2. Interoperability. The descriptions of different RIs will be compared and check the commonality of the operations, and validate the effectiveness of the reference model in realizing the interoperability between RIs. The development of the use case in the work package 4 will also be used as the scenario to test the reference model.
3. Application. The linking model and the reference model will be tested in the application planning systems to check the data, resource and infrastructure interoperability

Semantic linking model

The reference model will be used as an important input for task 3.4, namely the development of semantic linking model among the reference model, data and infrastructure. The linking model provides an information framework to glue different information models of resources and data. It couples the semantic descriptions of the data with the infrastructures and provides semantic interoperability between data and resources. It needs to address fault tolerance, optimization and scheduling of linked resources, while making a trade-off between fuzzy logic and full information. The linking model is part of the development effort of the reference model.

The linking model will take different aspects into considerations:

- The application (such as workflow) aspect captures the main characteristics of the application supported by the research infrastructure, including issues such as main flow patterns, quality of services, security and policies in user communities, and linking them to the descriptions of the data and infrastructures.
- The computing and data aspect focuses on operations and different data and meta data standards at different phase of data evolution (raw data, transfer, calibration, fusion etc.) and model them with linking of the data storing, accessing, delivery and etc. on (virtualized) e-Infrastructure.
- The Infrastructure aspect links the semantic model of the different layers of components in the physical infrastructure such as network elements and topologies, and also the monitoring information of the runtime status of the infrastructure. This part will enable the constraint solving of quality constraints to reserve and allocating resources for high level applications (processes).

5 REFERENCES

[1] W. Los, "Introduction to ENVRI and the workshop objectives," in *ENVRI Frascati Meeting 5-7 Feb 2013*, Presentation. Frascati, Italy, 2013.

[2] "Global Change: Towards global research infrastructures," *European Commission, Directorate-General For Research and Innovation*, 2012.

[3] S. Sorvari. "Envrionmental reseach in harmony," *International Innovation - Disseminating, science researhc and technology*. Dec. 2012 Page 28, 2012. Available: http://www.research-europe.com/magazine/ENVIRONMENT/2012-15/index.html

[4] ISO/IEC, "ISO/IEC 10746-1: Information technology--Open Distributed Processing--Reference model: Overview," *ISO/IEC Standard*, 1998.

[5] ISO/IEC, "ISO/IEC 10746-2: Information technology--Open Distributed Processing--Reference model: Foundations," *ISO/IEC Standard*, 2009.

[6] ISO/IEC, "ISO/IEC 10746-3: Information technology--Open Distributed Processing--Reference model: Architecture," *ISO/IEC Standard*, 2009.

[7] ISO/IEC, "ISO/IEC 10746-4: Information technology--Open Distributed Processing--Reference model: Architecture Semantics, "*ISO/IEC Standard*, 1998.

[8] OASIS, "Reference Model for Service Oriented Architecture 1.0," *OASIS Standard*, 2006.

[9] L. Candela, and A. Nardi, "The Digital Library Reference Model," *DL.org*, 2010.

[10] ISO/IEC, "Open System Interconnection (OSI), ISO/IEC 7498-1," *ISO/IEC Standard*, 1994.

[11] CCSDS, "Reference Model for an Open Archival Information System (OAIS)," *CCSDS Standard*, 2012.

[12] C. Atkinson, M. Gutheil, and K. Kiko, "On the Relationship of Ontologies and Models," *Lecture Notes in Informatics, Gesellschaft für Informatik, Bonn,* INI Proceedings, 1996.

[13] D. C. Schmidt, "Model-Driven Engineering," *IEEE Computer* vol. 39, 2006.

[14] N. F. Noy, and D. L. McGuinness, "Ontology Development 101: A Guide to Creating Your First Ontology," *Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880*, 2001.

[15] P. Tetlow, J. Z. Pan, D. Oberle, E. Wallace, M. Uschold, and E. Kendall, "Ontology Driven Architectures and Potential Uses of the Semantic Web in Systems and Software Engineering," *W3C Standard*, 2006.

[16] SEKE, "International Conference on Software Engineering (SEKE 2005)".

[17] VORTE, "International Workshop on Vocabularies, Ontologies and Rules for The Enterprise (VORTE 2005-2013)".

[18] MDSW, "The Model-Driven Semantic Web Workshop (MDSW 2004)".

[19] SWESE, "Workshop on Semantic Web Enabled Software Engineering (SWESE 2005-2007)".

[20] ONTOSE, "Workshop on Ontology, Conceptualizations and Epistemology of Software and Systems Engineering (ONTOSE 2005-2009)".

[21] WoMM, "Workshop on Meta-Modeling and Corresponding Tools (WoMM 2005)" .

[22] Kwaaitaal, M. Hoogeveen, and T. V. D. Weide, "A Reference Model for the Impact of Standardisation on Multimedia Database Management Systems," *Computer Standards & Interfaces*, vol. 16, pp. 45-54, 1994.

[23] OGC, "OGC Reference Model," *Open Geospatial Consortium*, OGC Standard, 2011.

[24] T. Uslander, "Reference Model for the ORCHESTRA Architecture (RM-OA) V2,", *Open Geospatial Consortium*, OGC Standard, 2007.

[25] V. Hernandez-Ernst, *et al.*, "LIFEWATCH. Deliverable 5.1.3: Data & Modelling Tool Structures -- Reference Model," *the EU LifeWatch consortium*, 2010.

[26] D. Hollingsworth, "The Workflow Reference Model," *the Workflow Management Coalition*, 1995.

[27] Mayk, and W. C. Regli, "Agent Systems Reference Model Release Version 1.0a," *US Army Communications and Electronics Command Research Development and Engineering Center (CERDEC)*, 2006.

[28] E. H. Chi, and J. T. Riedl, "An Operator Interaction Framework for Visualization Systems," *Symposium on Information Visualization (InfoVis '98)*, 1998.

[29] E. H. Chi, "A Taxonomy of Visualisation Techniques using the Data State Reference Model," *Proceedings of the IEEE Symposium on Information Visualization 2000 (InfoVis'00)*, 2000.

[30] N. Koch, and M. Wirsing, "The Munich Reference Model for Adaptive Hypermedia Applications," *in 2nd International Conference on Adaptive Hypermedia and Adaptive Web-based Systems, Proceedings. P. De Bra, P. Brusilovsky, and R. Conejo (eds.) LNCS 2347, ©Springer Verlag*, pp. 213-222, 2002.

[31] OMG, "Data Distribution Service for Real-time Systems Version 1.2," *OMG Standard*, 2007.

[32] OASIS, "Content Management Interoperability Services (CMIS) Version 1.0," *OASIS Stanard*, 2011.

[33] Hardisty, "WP3 Progress and Issues for Full Plenary with SAB, Wednesday 6th Feb 2013," in *ENVRI Frascati Meeting, 5-7 Feb 2013*, ed. Frascati, Italy, 2013

[34] R. Kahn and R. Wilensky, "A framework for distributed digital object services", *International Journal on Digital Libraries (2006) 6(2):115-123*, 2006.

[35] Barros, M. Dumas and P. Oaks, "A Critical Overview of the Web Services Choreography Description Language (WS_CDL)", *BPTrends*, Mar. 2005.

[36] L. Candela. "Data Use - Virtual Research Environments". In K. Ashley, C. Bizer, L. Candela, D. Fergusson, A. Gionis, M. Heikkurinen, E. Laure, D. Lopez, C. Meghini, P. Pagano, M. Parsons, S. Viglas, D. Vitlacil, and G. Weikum, *(ed.) Technological & Organisational Aspects of a Global Research Data Infrastructure - A view from experts, GRDI2020, 91-98*, 2012.

[37] P. F. Linington, Z. Milosevic, A. Tanaka, and A. Vallecillo, Ed., *Building Enterprise Systems with ODP*. CRC Press, 2012.

APPENDIXES

A. Terminology and Glossary

A.1 Acronyms and Abbreviations

CCSDS	Consultative Committee for Space Data Systems
CMIS	Content Management Interoperability Services
DDS	Data Distribution Service for Real-Time Systems
ENVRI	Environmental Research Infrastructure
ENVRI_RM	ENVRI Reference Model
ESFRI	European Strategy Forum on Research Infrastructures
ESFRI-ENV RI	ESFRI Environmental Research Infrastructure
GIS	Geographic Information System
IEC	International Electrotechnical Commission
ISO	International Organisation for Standardization
OAIS	Open Archival Information System
OASIS	Advancing Open standards for the Information Society
ODP	Open Distributed Processing
OGC	Open Geospatial Consortium
OMG	Object Management Group
ORCHESTRA	Open Architecture and Spatial Data Infrastructure for Risk Management
ORM	OGC Reference Model
OSI	Open Systems Interconnection
OWL	Web Ontology language
SOA	Service Oriented Architecture
SOA-RM	Reference Model for Service Oriented Architecture
RDF	Resource Description Framework
RM-OA	Reference Model for the ORCHESTRA Architecture
RM-ODP	Reference Model of Open Distributed Processing
UML	Unified Modelling Language
W3C	World Wide Web Consortium
UML4ODP	Unified Modelling Language For Open Distributed Processing

A.2 Terminology

Access Broker: Broker for facilitating data access requests.

Access Control: A functionality that approves or disapproves of access requests based on specified access policies.

Acquisition Service: An oversight service for an integrated data acquisition instrument network.

Active role: A active role is typically associated with a human actor.

Admin Gateway Host: A gateway service or equivalent construct used to interact with the infrastructure in an administrative capacity.

Authentication: A functionality that verifies a credential of a user.

Authentication Service: Security service responsible for the authentication of external agents making requests of infrastructure services.

Authorisation: A functionality that specifies access rights to resources.

Authorisation Service: Security service responsible for the authorisation of all requests made of infrastructure services by external agents.

Backup: A copy of computer data so it may be used to restore the original after a data loss event.

Behaviour : A behaviour of a community is a composition of actions performed by roles normally addressing separate business requirements.

Capacity Manager: An active role, which is a person who manage and ensure that the IT capacity meets current and future business requirements in a cost-effective manner.

Catalogue Service: An oversight service for metadata management with the data curation subsystem.

Citation Service: Access service for resolving citations used in external publications to refer to artefacts within a research infrastructure.

Community: A collaboration which consists of a set of *roles* agreeing their objective to achieve a stated business purpose.

Community Gateway Host: A gateway service or equivalent construct used by the scientific community to interact with the infrastructure.

Community Support Subsystem: A subsystem that provides functionalities to manage, control, and track users' activities and supports users to conduct their roles in the community.

Concept: Name and definition of the meaning of a thing. (abstract or real thing). Human readable definition by sentences, machine readable definition by relations to other concepts (machine readable sentences)

Conceptual Model: A collection of concepts, their attributes and their relations.

(persistent) Data: Term used as defined in ISO/IEC 10746-2. Data is the representations of information dealt by information systems and users thereof.

Data Access Subsystem: A subsystem that enables discovery and retrieval of data housed in data resources.

Data Acquisition Community. A community, which collects raw data and bring (streams of) measures into a system.

Data Acquisition Subsystem: A subsystem that collects raw data and brings the measures or data streams into a computational system.

Data Analysis: A functionality that inspects, cleans, transforms data, and provides data models with the goal of highlighting useful information, suggesting conclusions, and supporting decision making.

Data Assimilation: A functionality that combines observational data with output from a numerical model to produce an optimal estimate of the evolving state of the system.

Data Cataloguing: A functionality that associates a data object with one or more metadata objects which contain data descriptions.

Data Citation: A functionality that assigns an accurate, consistent and standardised reference to a data object, which can be cited in scientific publications.

Data Collection: A functionality that obtains digital values from a sensor instrument, associating consistent timestamps and necessary metadata.

Data Collector: An active role, which is a person who prepares and collects data. The purpose of data collection is to obtain information to keep on record, to make decisions about important issues, or to pass information on to others.

Data Consumer: Either an active or a passive role, which is an entity who receives and use the data.

Data Curation Community: A community, which curates the scientific data, maintains and archives them, and produces various data products with metadata.

Data Curation Subsystem: A subsystem that facilitates quality control and preservation of scientific data.

Data Curator: An active role, which is a person who verifies the quality of the data, preserve and maintain the data as a resource, and prepares various required data products.

Data Discovery and Access: A functionality that retrieves requested data from a data resource by using suitable search technology.

Data Export Service: An Oversight service for export of data from the data curation subsystem.

Data Extraction: A functionality that retrieves data out of (unstructured) data sources, including web pages, emails, documents, PDFs, scanned text, mainframe reports, and spool files.

SEVENTH FRAMEWORK PROGRAMME

Data Identification: A functionality that assigns (global) unique identifiers to data contents.

Data Import Service: An Oversight service for the import of new data into the data curation subsystem.

Data Mining: A functionality that supports the discovery of patterns in large data sets.

Data Originator: Either an active or a passive role, which provide the digital material to be made available for public access.

Data Processing Control: A functionality that initiates the calculation and manages the outputs to be returned to the client.

Data Processing Subsystem: A subsystem that aggregates the data from various resources and provides computational capabilities and capacities for conducting data analysis and scientific experiments.

Data Processor: Used by the data curation subsystem to perform simple processing of newly imported / exported data and invoke the data processing subsystem for more complex tasks.

Data Product Generation: A functionality that processes data against requirement specifications and standardised formats and descriptions.

Data Provenance: Information that traces the origins of data and records all state changes of data during their lifecycle and their movements between storages.

Data Provider: Either an active or a passive role, which is an entity providing the data to be used.

Data Publication: A functionality that provides clean, well-annotated, anonymity-preserving datasets in a suitable format, and by following specified data-publication and sharing policies to make the datasets publically accessible or to those who agree to certain conditions of use, and to individuals who meet certain professional criteria.

Data Publication Community: A community that assists the data publication, discovery and access.

(Data Publication) Repository: A passive role, which is a facility for the deposition of published data.

Data Quality Checking: A functionality that detects and corrects (or removes) corrupt, inconsistent or inaccurate records from data sets.

Data Query Service: An oversight service for the querying of data resources in the data curation subsystem.

Data Resource Host: A data store within the data curation subsystem.

Data Resource Registrar: Registration service for integrating new data resources into the data curation subsystem.

Data Service Provision Community: A community that provides various services, applications and software/tools to link, and recombine data and information in order to derive knowledge.

Data State: Term used as defined in ISO/IEC 10746-2. At a given instant in time, data state is the condition of an object that determines the set of all sequences of actions (or traces) in which the object can participate.

Data Storage & Preservation: A functionality that deposits (over long-term) the data and metadata or other supplementary data and methods according to specified policies, and makes them accessible on request.

Data Transmission: A functionality that transfers data over communication channel using specified network protocols.

Data Usage Community: A community who makes use of the data and service products, and transfers the knowledge into understanding.

Design of Measurement Model: A behaviour that designs the measurement or monitoring model based on scientific requirements.

Education or Trainee: An active role, a person, who makes use of the data and application services for education and training purposes.

ENVRI Reference Model: A common ontological framework and standards for the description and characterisation of computational and storage systems of ESFRI environmental research infrastructures.

Environmental Scientist: An active role, which is a person who conduct research or perform investigation for the purpose of identifying, abating, or eliminating sources of pollutants or hazards that affect either the environment or the health of the population. Using knowledge of various scientific disciplines, may collect, synthesize, study, report, and recommend action based on data derived from measurements or observations of air, food, soil, water, and other sources.

Execution Resource Host: Part of the execution platform provided by the data processing subsystem.

External Resource: A resource not within the infrastructure acting temporarily as a data source or destination for data export.

General Public, Media or Citizen (Scientist): An active role, a person, who is interested in understanding the knowledge delivered by an environmental science research infrastructure, or discovering and exploring the knowledgebase enabled by the research infrastructure.

Identification Service: Oversight service for identifier assignment and resolution.

Instrument Host: An instrument (integrated raw data source) in the data acquisition subsystem.

Mapping Rule: Configuration directives used for model-to-model transformation.

(Measurement Model) Designer: An active role, which is a person who design the measurements and monitoring models based on the requirements of environmental scientists.

SEVENTH FRAMEWORK PROGRAMME

Measurer: An active role, which is a person who determines the ratio of a physical quantity, such as a length, time, temperature etc., to a unit of measurement, such as the meter, second or degree Celsius.

Measurement Result: Quantitative determinations of magnitude, dimension and uncertainty to the outputs of observation instruments, sensors (including human observers) and sensor networks.

Metadata: Data about data, in scientific applications is used to describe, explain, locate, or make it easier to retrieve, use, or manage an information resource.

Metadata Harvesting: A functionality that (regularly) collects metadata (in agreed formats) from different sources.

Passive Role: A passive role is typically associated with a non-human actor.

PID generator: A passive role, a system which assigns persist global unique identifiers to a (set of) digital object.

PID registry: A passive role, which is an information system for registering PIDs.

Policy or Decision Maker: An active role, a person, who makes decisions based on the data evidences.

Process Control: A functionality that receives input status, applies a set of logic statements or control algorithms, and generates a set of analogue / digital outputs to change the logic states of devices.

Process Coordination Service: An oversight service for data processing tasks deployed on infrastructure execution resources.

Process Resource Registrar: Registration service for integrating new execution resources into the data processing subsystem.

Private Sector (Industry investor or consultant): An active role, a person, who makes use of the data and application service for predicting market so as to make business decision on producing related commercial products.

Provenance: The pathway of data generation from raw data to the actual state of data.

Quality Assessment (QA): Assessment of details of the data generation, including the check of the plausibility of the data. Usually the quality assessment is done by predefined checks on data and their generation process.

QA notation: Notation of the result of a Quality Assessment. This notation can be a nominal value out of a classification system up to a comprehensive (machine readable) description of the whole QA process.

Observer: An active role, which is a person who receives knowledge of the outside world through the senses, or records data using scientific instruments.

Reference Mode: A reference mode is an abstract framework for understanding significant relationships among the entities of some environment.

Resource Registration: A functionality that creates an entry in a resource registry and inserts resource object or a reference to a resource object in specified representations and semantics.

Role : A role in a community is a prescribing behaviour that can be performed any number of times concurrently or successively.

Service Consumer: Either an active or a passive role, which is an entity using the services provided.

Service Registry: A passive role, which is an information system for registering services.

Scientific Modelling and Simulation: A functionality that supports the generation of abstract, conceptual, graphical or mathematical models, and to run an instance of the model.

Scientist or Researcher: An active role, a person, who makes use of the data and application services to conduct scientific research.

(Scientific) Workflow Enactment: A specialisation of Workflow Enactment, which support of composition and execution a series of computational or data manipulation steps, or a workflow, in a scientific application. Important processes should be recorded for provenance purposes.

Semantic Annotation: link from a thing (single datum, data set, data container) to a concept within a conceptual model, enabling the discovery of the meaning of the thing by human and machines.

Semantic Mediator: A passive role, which is a system or middleware facilitating semantic mapping discovery and integration of heterogeneous data.

Service Provider: Either an active or a passive role, which is an entity providing the services to be used.

Sensor: A passive role, which is a converter that measures a physical quantity and converts it into a signal which can be read by an observer or by an (electronic) instrument.

Sensor network: A passive role, which is a network consists of distributed autonomous sensors to monitor physical or environmental conditions.

Specification of Scientific Measurements : The description of the scientific measurement model which specifies:

- what is measured;
- how it is measured;
- by whom it is measured; and
- what the temporal design is (single /multiple measurements / interval of measurement etc.)

Storage: A passive role, which is memory, components, devices and media that retain digital computer data used for computing for some interval of time.

Storage Administrator: An active role, which is a person who has the responsibilities to the design of data storage, tune queries, perform backup and recovery operations, raid mirrored arrays, making sure drive space is available for the network.

SEVENTH FRAMEWORK PROGRAMME

Subsystem: A subsystem is a set of capabilities that collectively are defined by a set of interfaces with corresponding operations that can be invoked by other subsystems. Subsystems are disjoint from each other.

Technician: An active role, which is a person who develop and deploy the sensor instruments, establishing and testing the sensor network, operating, maintaining, monitoring and repairing the observatory hardware.

Technologist or Engineer: An active role, a person, who develop and maintains the research infrastructure.

Unique Identifier (UID): With reference to a given (possibly implicit) set of objects, a unique identifier (UID) is any identifier which is guaranteed to be unique among all identifiers used for those objects and for a specific purpose.

User Behaviour Tracking: A behaviour enabled by a *Community Support System* that to track the *Users*. If the research infrastructure has identity management, authorisation mechanisms, accounting mechanisms, for example, a Data Access Subsystem is provided, then the Community Support System either include these or work well with them.

User Profile Management: A behaviour enabled by a *Community Support System* that to support persistent and mobile profiles, where profiles will include preferred interaction settings, preferred computational resource settings, and so on.

User Working Space Management: A behaviour enabled by a *Community Support System* that to support work spaces that allow data, document and code continuity between connection sessions and accessible from multiple sites or mobile smart devices.

User Working Relationships Management: A behaviour enabled by a *Community Support System* that to support a record of working relationships, (virtual) group memberships and friends.

User Group Work Supporting: A behaviour enabled by a *Community Support System* that to support controlled sharing, collaborative work and publication of results, with persistent and externally citable PIDs.

Upload Broker: Broker for facilitating data upload requests from external contributors.

B. Common Requirements of ENVRI Research Infrastructures

From a pre-study of the ENVRI Research Infrastructures, a set of functionalities commonly provided by those research infrastructure have be identified, which is listed as follow highlighted with the minimal set of core functionalities**.**

A	**Data Acquisition Subsystem**	
No	**Functions**	**Definitions**
A.1	Instrument Integration	A functionality that creates, edits and deletes a sensor.
A.2	Instrument Configuration	A functionality that sets-up a sensor or a sensor network.
A.3	Instrument Calibration	A functionality that controls and records the process of aligning or testing a sensor against dependable standards or specified verification processes.
A.4	Instrument Access	A functionality that reads and/or updates the state of a sensor.
A.5	Configuration Logging	A functionality that collects configuration information or (run-time) messages from a sensor (or a sensor network) and output into log files or specified media which can be used by routine troubleshooting and in incident handling.
A.6	Instrument Monitoring	A functionality that checks the state of a sensor or a sensor network which can be done periodically or when triggered by events.
A.7	(Parameter) Visualisation	A functionality that outputs the values of parameters and measured variables a display device.
A.8	*(Real-Time) (Parameter/Data) Visualisation*	A specialisation of (Parameter) Visualisation which is subject to a real-time constraint.
A.9	Process Control	An interface that provide operations to receive input status, apply a set of logic statements or control algorithms, and generate a set of analogy and digital outputs to change the logic states of devices.
A.10	Data Collection	An interface that provides operations to obtain digital values from a sensor instrument, associating consistent timestamps and necessary metadata.
A.11	*(Real-Time) Data Collection*	A specialisation of Data Collection which is subject to a real-time constraint.
A.12	Data Sampling	An interface that provides operations to select a subset of individuals from within a statistical population to estimate characteristics of the whole population.
A.13	Noise Reduction	An interface that provides operations to remove noise from scientific data.
A.14	Data Transmission	A interface that provides operations to transfer data over communication channel using specified network protocols.
A.15	*(Real-Time) Data Transmission*	A specialisation of Data Transmission which handles data streams using specified real-time transport protocols.

A.16	Data Transmission Monitoring	An interface that provides operations to check and report the status of data transferring process against specified performance criteria.
B	**Data Curation Subsystem**	
No	**Functions**	**Definitions**
B.1	Data Quality Checking	An interface that provides operations to detect and correct (or remove) corrupt, inconsistent or inaccurate records from data sets.
B.2	Data Quality Verification	An interface that provides operations to support manual quality checking.
B.3	Data Identification	An interface that provides operations to assign (global) unique identifiers to data contents.
B.4	Data Cataloguing	An interface that provides operations to associate a data object with one or more metadata objects which contain data descriptions.
B.5	Data Product Generation	An interface that provides operations to process data against requirement specifications and standardised formats and descriptions.
B.6	Data Versioning	A interface that provides operations to assign a new version to each state change of data, allow to add and update some metadata descriptions for each version, and allow to select, access or delete a version of data.
B.7	Workflow Enactment	An interface that provide operations or services to interprets predefined process descriptions and control the instantiation of processes and sequencing of activities, adding work items to the work lists and invoking application tools as necessary.
B.8	Data Storage & Preservation	An interface that provides operations to deposit (over long-term) the data and metadata or other supplementary data and methods according to specified policies, and make them accessible on request.
B.9	Data Replication	An interface that provides operation to create, delete and maintain the consistency of copies of a data set on multiple storage devices.
B.10	Replica Synchronisation	An interface that provides operations to export a packet of data from on replica, transport it to one or more other replicas and to import and apply the changes in the packet to an existing replica.
C	**Data Access Subsystem**	
No	**Functions**	**Definitions**
C.1	Access Control	An interface that provides operations to approve or disapprove of access requests based on specified access policies.
C.2	Resources Annotation	An interface that provides operations to create, change or delete a note that reading any form of text, and to associate them with a computational object.

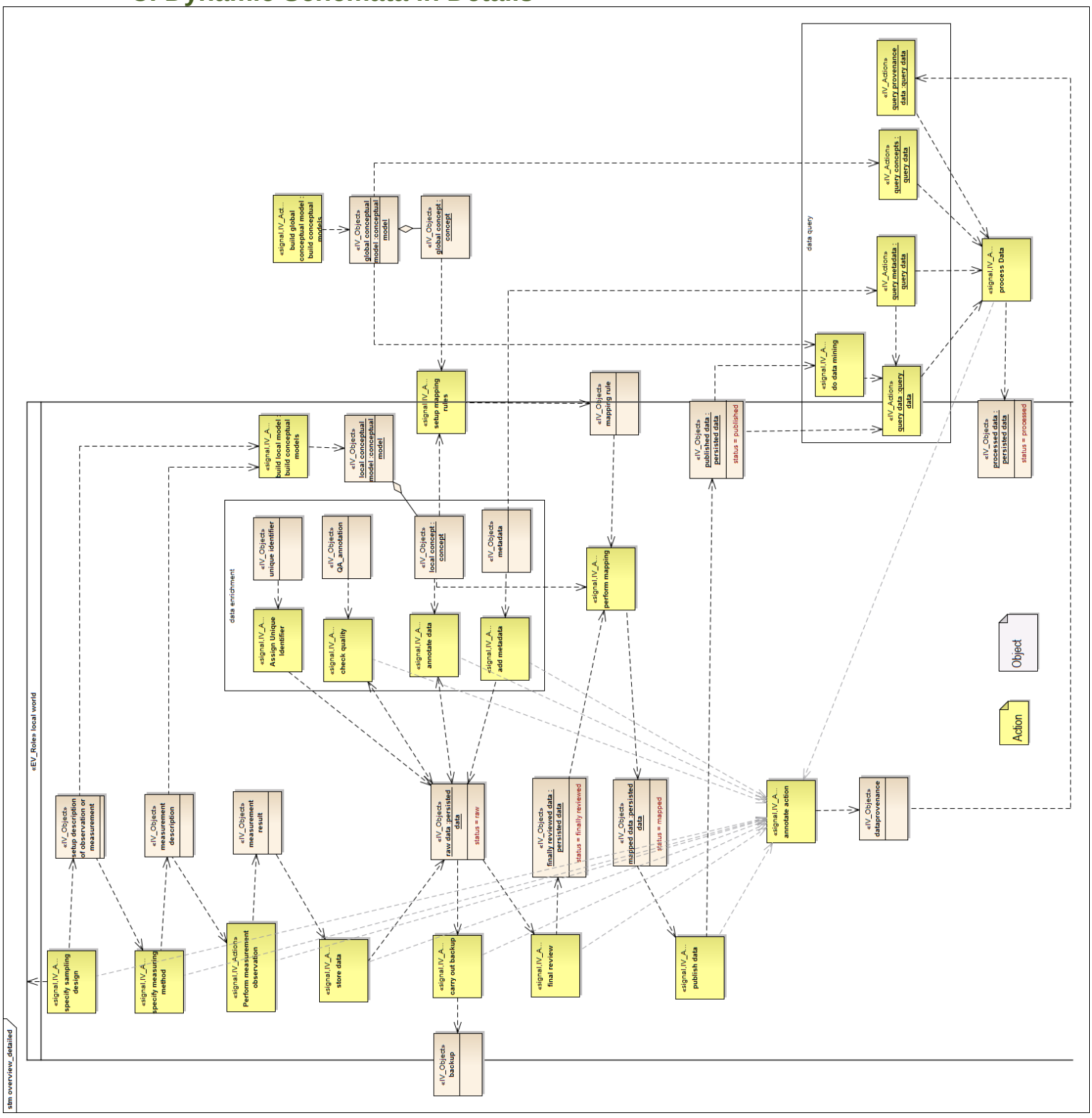
SEVENTH FRAMEWORK PROGRAMME

C.3	*(Data) Annotation*	A specialisation of Resource Annotation which allows to associate an annotation to a data object.
C.4	Metadata Harvesting	An interface that provides operations to (regularly) collect metadata (in agreed formats) from different sources.
C.5	Resource Registration	An interface that provides operations to create an entry in a resource registry and insert resource object or a reference to a resource object in specified representations and semantics.
C.6	*(Metadata) Registration*	A specialisation of Resource Registration, which registers a metadata object in a metadata registry.
C.7	*(Identifier) Registration*	A specialisation of Resource Registration, which registers an identifier object in an identifier registry.
C.8	*(Sensor) Registration*	A specialisation of Resource Registration which registers a sensor object to a sensor registry.
C.9	Data Conversion	An interface that provides operations to convert data from one format to another format.
C.10	Data Compression	An interface that provides operations to encode information using reduced bits by identifying and eliminating statistical redundancy.
C.11	Data Publication	An interface that provides operations to provide clean, well-annotated, anonymity-preserving datasets in a suitable format, and by following specified data-publication and sharing policies to make the datasets publically accessible or to those who agree to certain conditions of use, and to individuals who meet certain professional criteria.
C.12	Data Citation	An interface that provides operations to assign an accurate, consistent and standardised reference to a data object, which can be cited in scientific publications.
C.13	Semantic Harmonisation	An interface that provides operations to unify similar data (knowledge) models based on the consensus of collaborative domain experts to achieve better data (knowledge) reuse and semantic interoperability.
C.14	Data Discovery and Access	An interface that provides operations to retrieve requested data from a data resource by using suitable search technology.
C.15	Data Visualisation	An interface that provides operations to display visual representations of data.
D	**Data Processing Subsystem**	
No	**Functions**	**Definitions**
D.1	Data Assimilation	An interface that provides operations to combine observational data with output from a numerical model to produce an optimal estimate of the evolving state of the system.
D.2	Data Analysis	An interface that provides operations to inspect, clean, transform data, and to provide data models with the goal of highlighting useful information, suggesting conclusions, and supporting decision making.

D.3	Data Mining	An interface that provides operations to support the discovery of patterns in large data sets.
D.4	Data Extraction	A interface that provides operations to retrieve data out of (unstructured) data sources, including web pages ,emails, documents, PDFs, scanned text, mainframe reports, and spool files.
D.5	Scientific Modelling and Simulation	An interface that provides operations to support of the generation of abstract, conceptual, graphical or mathematical models, and to run an instance of the model.
D.6	*(Scientific) Workflow Enactment*	A specialisation of Workflow Enactment, which support of composition and execution a series of computational or data manipulation steps, or a workflow, in a scientific application. Important processes should be recorded for provenance purposes.
D.7	(Scientific) Visualisation	An interface that provides operations to graphically illustrate scientific data to enable scientists to understand, illustrate and gain insight from their data.
D.8	Service Naming	An interface that provides operations to encapsulate the implemented name policy for service instances in a service network.
D.9	Data Processing	An interface that provides operations to initiate the calculation and manage the outputs to be returned to the client.
D.10	Data Processing Monitoring	An interface that provides operations to check the states of a running service instance.
E	**Community Support Subsystem**	
No	**Functions**	**Definitions**
E.1	Authentication	An interface that provides operations to verify a credential of a user.
E.2	Authorisation	An interface that provides operations to specify access rights to resources.
E.3	Accounting	An interface that provides operation to measure the resources a user consumes during access for the purpose of capacity and trend analysis, and cost allocation.
E.4	*(User) Registration*	A specialisation of Resource Registration which registers a user to a user registry.
E.5	Instant Messaging	An interface that provides operation for quick transmission of text-based messages from sender to receiver.
E.6	(Interactive) Visualisation	An interface that provides operations to enable users to control of some aspect of the visual representations of information.
E.7	Event Notification	An interface that provide operations to deliver message triggered by predefined events.

SEVENTH FRAMEWORK PROGRAMME

C. Dynamic Schemata in Details



SEVENTH FRAMEWORK PROGRAMME

Before a measurement or observation can be started the design (or setup) must be defined, including the working hypothesis and scientific question, method of the selection of sites (stratified / random), necessary precision of the observation or measurement, boundary conditions, etc. For correctly using the resulting data, detailed information about that process and its parameters have to be available for people processing the data. (e.g. if a stratified selection of sites according to parameter A is done, the resulting value of parameter A cannot be evaluated in the same way as other results)

After defining the overall design of measurements or observations, the measurement method, complying with the design, including devices which should be used, standards / protocols which should be followed, and other details have to be specified. Information of that process and the parameters resulting of the process have to be stored in order to guarantee correct interpretation of the resulting data. (e.g. when you want to model a dependency of parameter B of a parallel measured wind velocity, the limit of detection of the used anemometer influences the range of values of possible assertions).

When the measurement or observation method is defined, it can be carried out, producing measurement results. The handling of those results, all the actions done, to store the data are pulled together in the action "store data". (This action can be very simple when using a measurement device, which periodically sends the data to the data management system, but this can also be a sophisticated harvesting process or e.g. in case of biodiversity observations a process done by humans). The storage process is the first step in the life cycle of data that makes data accessible in digital form and are persisted.

As soon as data are available for IT purposes a backup can be made, independently of the state of the persisted data. This can be done locally or remote, done by the data owners or by dedicated data curation centers. At any status of the data can be processed for QA-assessments, for readjustment of the measurement or observation design and a lot of other reasons. Evaluations, which lead to answers of the scientific question, however, are usually done on data with a certain status - the status "finally reviewed".

Raw data can get enriched, which means that additional information can be added to them:
They can get a unique identifier, necessary for unambiguous identification of data (allowing to identify copies), and resolution within the data provenance.

They can undergo a QA process, checking their plausibility, correcting noise and several other processes, adding the information about that process to the data.

Metadata might be linked to the data, either by application of a standard metadata schema or by following a proprietaries metadata description.

semantic annotation can be added linking the data with their meaning. this semantic annotation can reach from annotation of units over annotation about used taxonomic lists to pointers to concepts in ontologies, describing the background of the measurement or observation in a machine and human readable form.

SEVENTH FRAMEWORK PROGRAMME

Making data accessible for users outside the Environment of the data owner at least needs two steps: 1) Mapping the data to the "global" semantics, the semantics the data owner shares with the data user. 2) Publish the data. Mapping data to global semantics may include simple conversions like conversions of units but also need more sophisticated transformations like transformations of code lists and other descriptions like the setup descriptions, measurement descriptions, and data provenance. "global" and "local" are usual, but a little bit confusing terms. There is nothing like a conceptual model for the whole world, which might be expected when we talk about a global conceptual model. A conceptual model (even if it is just a list of used units) is always just valid for a certain community. Whenever the community using some data, is widened, this widened community may have its new conceptual model. The case that two communities have the same model is a very rare luck. The smaller community has the so called "local" conceptual model and the larger community the so called "global".

It is important to know about published data, whether those data have a status: "finally reviewed" and what such a status means. It can mean, that those data will never change again, the optimum for the outside user. But it might also mean, that only under certain circumstances those data will be changed. In this case it is important to know what "certain circumstances" means. And additionally it is important to know, how and where the used semantics are described. A resolvable pointer to them, of course is the solution which can be handled most easily.

All the steps within the life cycle of data can be stored as data provenance, containing at least information about the used objects, the produced objects and the applied action. There are two important use cases for data provenance: 1.) citation of data and all the actors involved in the production of the data. 2.) correct interpretation of the data.

Data can be made directly accessible or indirectly in two or more steps. Direct one step access means, that you send a data request to a data server (query data) and get the data or an error message as answer. Indirect access or two step access means, that you first access metadata (query metadata) , search for a probably fitting data set and then query the data. Those two steps can be extended to more than two steps, when intermediate steps are involved. The two or more step approach is often used for data, which are not open, making metadata open but data not open. For questions touching several datasets and/or filtering the data (like e.g. give me all NOx air measurement where O3 exceeds a level of Y ppb) this two-step approach can be seen as a high barrier.

SEVENTH FRAMEWORK PROGRAMME