e-science and technology infrastructure for biodiversity data and observatories

Deliverable 5.1.3

Data & Modelling Tool Structures

- Reference Model –

Fraunhofer IAIS
Cardiff University

Work Package 5a Construction plan strategy

Life Watch
e-Science and Technology infrastructures
for biodiversity data and observatories

EU Seventh Framework Programme (FP7) Infrastructures (INFRA-2007-2.2-01)

Contract Number 211372

Security (distribution level)	Public
Actual date of delivery of v0.5	2010-01-08
Deliverable number	5.1.3
Deliverable name	LifeWatch Technical Construction Plan –Reference Model
Type	Report
Status & version	Working Document v0.5
Number of pages	230
WP contributing to the deliverable	5a
Task responsible	Fraunhofer IAIS
Other contributors	Cardiff University
Author(s)	Vera Hernandez-Ernst Axel Poigné Jon Giddy Alex Hardisty With contributions by W. Berendson H. Schentz Angie Voss

Revision History

Version	Description	Changed	Date	Done by
V0.0.1	First draft, edited on Google Docs	all	January 2009	V. Hernández A. Poigné A. Hardisty J. Giddy
V0.0.2	Reedited in Word	all	March 2009	V. Hernández A. Poigné
V0.1	1st draft for internal review by Work Package 5	all	2009-02-13	V. Hernández A. Poigné A. Hardisty J. Giddy
V0.2	1st public draft	all Reformatted (Major parts to Appendices)	2009-04-03	V. Hernández A. Poigné A. Hardisty J. Giddy With contribution by W. Berendson H. Schentz
V0.2.1	Reformatted	all	2009-06-22	A. Poigné
V0.3	2nd draft	Preamble: changed Section 3.9: added Section 6: changed Section 7: new Appendix H: new	2009-10-31	V. Hernández A. Poigné
V0.4	2nd public draft	Section 7: extended Section 6: changed Section 8: new	2009-12-07	V. Hernández A. Poigné
V0.5	3rd draft	Section 3.4 + 3.5 changed Section 8: expanded Appendix D to “Show Cases” Appendix G “Workflow Example” to Status Report Appendix H “SOA versus ROA” to Status Report Other minor changes		V. Hernández A. Poigné

Blank Page

Notices of Copyrights, Modifications, and Acknowledgements

Open Geospatial Consortium, Inc. License Agreement

BEFORE YOU CLICK ON THE ACCEPT BUTTON AT THE END OF THIS DOCUMENT, CAREFULLY READ ALL THE TERMS AND CONDITIONS OF THIS AGREEMENT. BY CLICKING ON THE ACCEPT BUTTON, YOU ARE CONSENTING TO BE BOUND BY AND ARE BECOMING A PARTY TO THIS AGREEMENT. IF YOU DO NOT AGREE TO ALL OF THE TERMS OF THIS AGREEMENT, CLICK THE "DO NOT ACCEPT" BUTTON AND DO NOT DOWNLOAD THIS INTELLECTUAL PROPERTY.

Readers of this document are requested to submit to Open Geospatial Consortium, Inc. ("Licensor"), with their comments, notification of any relevant patent rights or other intellectual property rights of which they may be aware which might be infringed by any use of the copyrighted material that follows (the "Intellectual Property"), as appropriate, and to provide supporting documentation.

Open Geospatial Consortium Inc., OGC, OPENGIS, and the OGC CERTIFIED COMPLIANT logo are registered trademarks or trademarks of Licensor in the United States and in other countries.

Permission is hereby granted, free of charge and subject to the terms set forth below, to any person obtaining a copy of this Intellectual Property and any associated documentation, to deal in the Intellectual Property without restriction (except as set forth below), including without limitation the rights to implement, use, copy, modify, merge, publish, distribute, and/or sublicense copies of the Intellectual Property, and to permit persons to whom the Intellectual Property is furnished to do so, provided that all copyright notices on the intellectual property are retained intact and that each person to whom the Intellectual Property is furnished agrees to the terms of this Agreement.

If you modify the Intellectual Property, all copies of the modified Intellectual Property must include, in addition to the above copyright notice, a notice that the Intellectual Property includes modifications that have not been approved or adopted by LICENSOR.

THIS LICENSE IS A COPYRIGHT LICENSE ONLY, AND DOES NOT CONVEY ANY RIGHTS UNDER ANY PATENTS THAT MAY BE IN FORCE ANYWHERE IN THE WORLD.

<end of OGC License Agreement>

OASIS Copyright Notice

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

Table of Content

List of Figures

List of Tables

Preamble

A Reference Model is an abstract framework for understanding the significant relations among the entities of the subject of concern. Its purpose is to provide a common conceptual framework defining the key characteristics that can be used consistently across different implementations. A Reference Model should be distinguished from a Reference Architecture that serves as an abstract model from which concrete architectures can be derived. The Reference Architecture focuses on the components of a system and their relationships, introducing concepts and architectural elements as needed in order to fulfil core requirements of the system to be constructed, but avoiding reliance of specific technologies.

The LifeWatch Reference Model may be considered as an intermediate between Reference Model and Reference Architecture as discussed above. It provides a common conceptual framework as well as it defines a number of components and architectural concepts as a basis for the future LifeWatch Architecture. It is neither a blueprint nor does it define a technological mapping, but identifies some key aspects and components that should be present in the final implementation of the LifeWatch System.

The LifeWatch Reference Model is considered being a "living document" that is going to evolve during the lifetime of the LifeWatch project. Its intention is, at every time, to represent a common view of the ICT dimension of all those involved and contributing to the LifeWatch project and to provide guidelines for the construction and management process.

The LifeWatch Reference Model should not be considered as a rigorous scheme to be enforced by the program management but rather as a document that helps to improve interoperability and to which services provided may conform to different degrees. Of course, the project should aim for the highest degree of conformance possible, in particular for core components.

According to the insights gained at each stage of the project, the Reference Model may/should be adapted, extended, more focussed, re-factored, or even rewritten. This needs efforts and expenses to be spent. The authors believe that these are well spent since discussion and adaptation of the Reference Model should provide interoperability between the components of the system and, even more importantly, cohesion of the aims and procedures of all those involved. Controlling the development of the Reference Model and controlling the correlation between the LifeWatch Reference Model and the actual development work is considered as an important task of the project management.

The present version of the LifeWatch Reference Model reflects the insights gained in the preparatory phase. These insights are not backed by actual construction work but constitutes a compilation of best practice experience gained in other projects as well as an extensive study of the literature as reflected by the Status Report on Infrastructures for Biodiversity Research1. Basically, all the concepts and the terminology used in this document are covered in the status report. This document refrains from giving explicit references not to clutter up the text with too many references.

Editor's Note: *Texts in blue italics (italics only when viewing in black and white) are placeholders for our own comments on the procedures or on future developments. These are generally marked as "Editor's Note".*

[1] LifeWatch Working Document "Infrastructures for Biodiversity Research – Status Report, May 2009

Blank Page

1 Introduction

1.1 General

"At the heart of biodiversity informatics there is the dream of a unified infrastructure where data and analysis services all around the world are seamlessly accessible and integrated."2

This document provides the first draft of a Reference Model for the LifeWatch ICT infrastructure as a service-oriented architecture. The Reference Model is based on the ORCHESTRA Reference Model (see 0 for a summary) that builds on standards for distributed processing and geospatial computing. The Reference Model includes guidelines for the specification and implementation of the LifeWatch ICT infrastructure as well as defining a number of generic information models and services.

The present draft mainly focuses on the guidelines for specification and implementation.

Note: The document addresses two audiences: the biodiversity community and the biodiversity informatics community, including those involved in specifying and implementing the LifeWatch Infrastructure. Much of the material will mainly address the latter community in that the document tries to set structural boundary conditions for an engineering project that involves many teams from many countries funded by many funding agencies for a long time with technologies certain to change over that time. These boundary conditions are defined in terms of a Reference Model for the LifeWatch infrastructure, conformance with which will classify what is and is not within LifeWatch.

The most important parts for the biodiversity community are this section, Section 3, and Section 4, and, may be, the first subsection of each of the other sections. For all other readers, we recommend reading of all sections. However, this is only part of the task as we only give an outline of many details of the ORCHESTRA ICT infrastructure that our approach very much relies on. System specification and software construction demands for a deeper understanding of the ORCHESTRA approach (http://www.eu-orchestra.org/) and many of the standards involved.

1.2 Requirements for the specification framework

The LifeWatch ICT Infrastructure (or infrastructure for short) shall be a distributed system of nodes that provide access to and processing of biodiversity data from a variety of sources through common open interfaces for several decades.

Whilst it is difficult to predict future developments, even on the conceptual level, there are common principles such as reusability, modularity, portability, interoperability, discoverability, and compliance with standards. The LifeWatch infrastructure shall:

- Rigorously use proven concepts and standards to avoid dependence on vendor specific solutions and to maintain the freedom to use all the emerging solutions based on these standards;
- Rigorously use proven concepts and standards to avoid dependence on vendor specific solutions and to maintain the freedom to use all the emerging solutions based on these standards;
- Comply with the INSPIRE Directive and Implementation Guidelines for spatial data infrastructures in Europe3;
- Consist of loosely coupled components which can be interconnected using mediation;

---

[2] Citation from the report of the BioGeoSDI workshop, Brazil, April 2007 (http://wiki.tdwg.org/twiki/pub/Geospatial/InteroperabilityWorkshop1/BioGeoSDIreport.pdf)

[3] http://inspire.jrc.ec.europa.eu/

- Be independent of specific technologies to accommodate future changes of technology;
- Support an evolutionary style of development;
- Be loosely coupled with external systems;
- Be designed in a flexible, generic, and adaptable way for usage across different thematic areas and contexts; and,
- Shall implement and deploy infrastructure using established techniques that guarantee rapid availability of components, whilst in parallel carrying out experimental research into cutting-edge technologies in selected areas to ensure adoption of new approaches, contributing to European Research Area informatics development.

At present, Service Oriented Architecture (SOA) appears to provide a solid conceptual basis for supporting these principles. Services constitute a practical mechanism for interoperability across the borders of institutions. Workflows are a common conceptual paradigm for specifying chains of services. Hence, the specification of the LifeWatch infrastructure is based on the assumptions that:

- Functionality is broken into component services based on the principles of Service Oriented Architecture;
- Semantic Services provide uniform semantically defined interfaces to a set of operations, enabling syntactic and semantic interoperability between and substitution of components; and,
- Workflows are used for the chaining of operations from multiple distributed services in order to perform specific user tasks.

Exposure of the semantics at the services interfaces and computing across those semantics, in conjunction with a rigorous adherence to accepted standards and the promotion of unified meta-models tends to lead towards an improved possibility of achieving greater interoperability. Hence, one may consider our approach as 'Semantic Service Oriented Architecture'. For convenience, we rely on the abbreviation 'SOA'.

Similar requirements and assumptions in the European "ORCHESTRA" Integrated Project4led to a Reference Model (RM_OA) as a generic specification framework for geospatial service-oriented architectures and service networks. ORCHESTRA is committed to principles for Open Distributed Processing (ODP) set by ISO/IEC 10746 and ISO 19119 and has achieved "best practice" status within the OGC consortium5. The ORCHESTRA Reference Model extends the OGC Reference Model, which in turn is based on the Reference Model for Open Distributed Processing (ISO/IEC 10746) (ODP) (Figure 1 below).

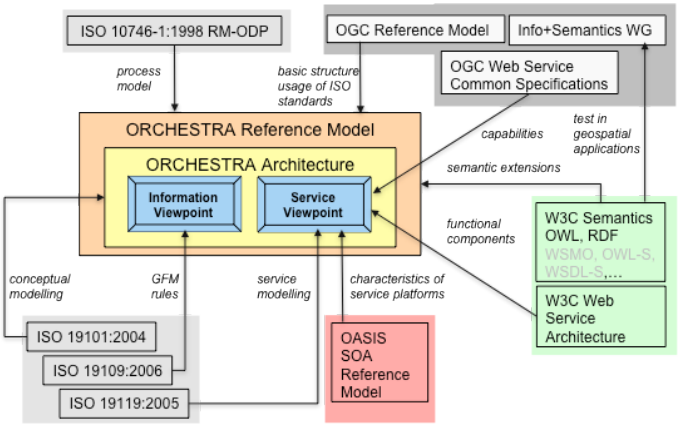


Figure 1: Relating the ORCHESTRA architecture to standards

[4] http://www.eu-orchestra.org/

[5] http://www.opengeospatial.org/standards/bp

LifeWatch intends to basis its specification around the ORCHESTRA approach and to maintain compatibility to the greatest extent possible, recognising also the need for specific extensions to support capabilities specific to the biodiversity domain.

0 of the present document provides a summary of the ORCHESTRA approach.

Further, the ORCHESTRA Architecture has been and is successfully used in European projects such as:

- SANY Sensors Anywhere6: The project focuses on interoperability of in-situ sensors and sensor networks and will provide a service-oriented architecture for environmental sensor networks. Applications are in the area of risk analysis of, for instance, air and water pollution; and,
- The GEOSS, INSPIRE and GMES an Action in Support (GIGAS)7 promotes the coherent and interoperable development of the GMES, INSPIRE and GEOSS initiatives through their concerted adoption of standards, protocols, and open architectures.

The maturity of the ORCHESTRA Reference Model, its conformance with international standards, its coverage of requirements similar to those of LifeWatch, as well as its adoption by major European initiatives are the reasons to adopt it for LifeWatch. Although environmental risk management is the application area in ORCHESTRA, the reference model provides means to deliver sustainable, reusable, and independent concepts and components. This permits three degrees of freedom, which give room for different scopes of individual platforms and the transition to new technologies in the long lifespan of the LifeWatch infrastructure. The degrees of freedom concern:

- Technical capabilities: A typical workflow moves from the discovery of data to the download of data, its transformation, and integration, the processing of data and the presentation of the results. Especially in the beginning of LifeWatch, some nodes may focus their effort on a few steps, such as discovery, download, and viewing of data. The LifeWatch Reference Model (LRM) is modular and allows the nodes to progressively expand their technical capabilities.
- Thematic extensions: Biodiversity is a multidisciplinary research area. The domains of the tasks to be performed with the infrastructure may expand from the pure analysis of occurrence records to, for example, analyses, modelling, and predictions that include ecological, genetic and phylogenetic data. The scope of the tasks may vary from platform to platform. The LifeWatch Reference Model will define a common basic application area and give rules for thematic extensions. One consequence of thematic extensibility is semantic extensibility allowing new, derived, and refined concepts.
- Technology: The Reference Model distinguishes between an abstract architecture and the concrete platform of the system. During the lifespan of LifeWatch, it may be necessary to switch to new platform technologies. Such a transition is simplified because old and new platform must map to the same abstract architecture. Isolating the implementation issues from the ICT perspective allows the LifeWatch programme to cope with and adapt to the rapid rate of technological change characteristic of the ICT sector.

1.3 The ORCHESTRA Reference Model and its extension for LifeWatch

ORCHESTRA promotes an incremental, iterative approach for the analysis and design phases. It distinguishes between an abstract service platform specified independently of any middleware technology and a concrete service platform that is implemented on a specific middleware - the Service Network (Figure 2 below).

---

6 IST FP6 Integrated Project, http://www.sany-ip.eu/

7 http://thegigasforum.eu/

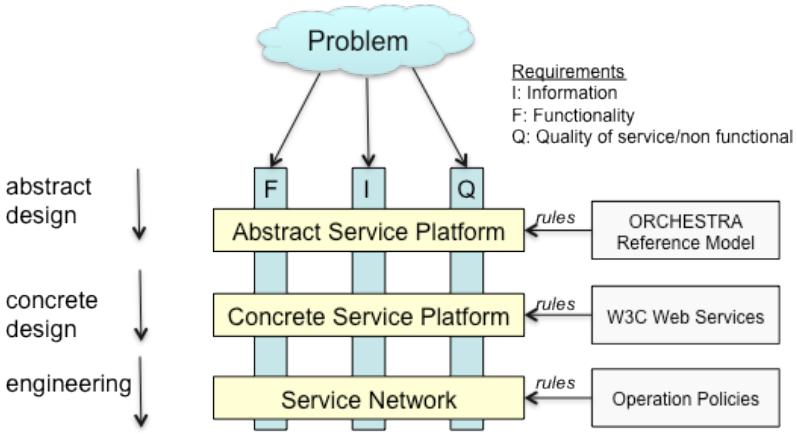


Figure 2: Abstract and concrete service platforms

In detail, the development phases considered are:

1. The analysis phase, resulting in requirements that constitute the "enterprise viewpoint".
2. The abstract design phase leads to platform-neutral specifications following the ORCHESTRA rules for the abstract service platform. They define the informational requirements (information model), functional requirements (abstract service specifications), and non-functional requirements (specification of the quality of service (QoS)) of LifeWatch.
3. The concrete design phase specifies a concrete service platform according to the rules of the Reference Model. In the current ORCHESTRA project, this is the ORCHESTRA Web Services platform consisting of the rules of the W3C Web services and a profile of the Geography Markup Language (GML) as the current mainstream service platform technologies for geospatial applications.
4. The engineering phase maps the platform-specific components to the abstract specifications and organises the concrete components into service networks taking into account the QoS requirements and translating them into operational policies.

Separating the abstract and the concrete design phases allows LifeWatch to cope with and adapt to the rapid rate of technological change characteristic of the ICT sector. We are not forced at the present stage of preparation to make detailed technology decisions that may well be obsolete by the time the construction phase commences. It also allows us, over the course of time, to evolve to new (lower cost, more efficient) technologies without necessarily affecting the capabilities offered to users.

The Reference Model for the ORCHESTRA Architecture (RM-OA) provides a number of predefined (generic) domain independent service types and rules for how to specify information models and services for building applications for a particular domain or task. A substantial number of elements from the ORCHESTRA Reference Model can be reused directly for LifeWatch:

- Rules and guidance about how to specify information and service models. In ORCHESTRA these rules are formally defined in a Reference Model for information and for services;
- Basic re-usable specification units for information models (e.g. pre-defined feature types, core ontology) and service models (e.g. re-usable interfaces);
- Textual and document templates that guide the specification of services; and,
- A series of textual descriptions and formal specifications of services (or, in short, architecture services8) that are application- and technology independent.

[8] 'Architecture Services in ORCHESTRA terminology. 'Basic Services' in LifeWatch terminology.

Using the rules of the ORCHESTRA Reference Model, LifeWatch will adapt and extend it in two directions:

- LifeWatch eventually refines and extends the ORCHESTRA Reference Model by introducing new generic information models, services and rules (the LifeWatch Reference Model), and
- LifeWatch defines some 'Base Application Architecture' with common biodiversity-oriented information models and services.

The LifeWatch Reference Model will define the level of conformance that all LifeWatch approved services and tools will have to comply with. A set of test cases and tools will be provided to support (semi-) automatic conformance testing.

The specification and design process began with the preparatory phase of the LifeWatch programme. Significant analysis is taking place in the preparatory phase. This will continue into the construction phase of the LifeWatch programme. In the preparatory phase, the specification and design focuses on:

1. Analysis:
    - Scope, goals, and policies for the ICT-infrastructure, derived from the LifeWatch Masterplan;
    - "Show cases" illustrating biodiversity themes and the supporting "biodiversity research capabilities" to make them possible. These in turn give rise to requirements for technical capabilities; and,
    - From the assessment of the state-of-the-art recommendations for the concrete platform, basic application, and thematic architecture, some core ontology, and the road map are derived.
2. Abstract design: In the preparatory phase the LifeWatch Reference Model will be detailed and architectural services, information models and a semantic framework (e.g. ontologies) for the Base LifeWatch Application architecture will be specified. Thematic extensions will be added and the Base LifeWatch Architecture will be extended as necessary during the construction phase.
3. Concrete design: In the preparatory phase, engineering and technology options are discussed and recommendations are given. The concrete platform specification will be fully detailed in the LifeWatch construction phase. It will grow in parallel with the abstract application architectures. The most important decision here concerns the service technology and consistency with the INSPIRE implementation rules.
4. Engineering phase: It will take place in the construction phase of LifeWatch. Here the mapping of the concrete to the abstract design will be defined and policies e.g. for naming, access control and service administration will be given.

1.4 Scope and structure of this document

This document elaborates the Reference Model for the LifeWatch Architecture. The present version will give a first sketch of the LifeWatch architecture and its components. This sketch will serve as background for the state-of-the-art report to come[9] which will analyse the projects, services, and tools related to biodiversity and ecological research, the goal being to identify those entities which may directly or with some amendments contribute to or be integrated into the LifeWatch infrastructure.

Later versions of the present document will extend this framework to the point of a blueprint. This blueprint will set the basis for developing a roadmap elaborating how available infrastructures could be enhanced to support the required functionalities within LifeWatch construction time scale. The blueprint together with the road map will constitute Deliverable 5.3b, "Final report on the Technical Construction

[9] Deliverable D5a.1 of the LifeWatch preparatory project – available soon.

Plan", which will then be a part of Deliverable 2.3, "Total construction plan", and in a reduced format be part of the LifeWatch Masterplan.

Revisions and refinement of this document will take place during the construction phase based on insights gained by following a middle-out approach where the stipulations made serve as a guideline for integrating new services while new services to be integrated may have an impact on the development of such a reference model.

The structure of the document follows the Reference Model of Open Distributed Processing [(ISO/IEC 10746), which defines different viewpoints for system specification, each of which addresses a different kind of actor. These viewpoints were also adopted in the OpenGIS Service Architecture (ISO 19119) and in the ORCHESTRA Reference Model:

- Section 3 presents the Enterprise Viewpoint capturing requirements;
- Section 4 gives a general overview of scope and structure of the LifeWatch architecture, all its components, and its implementation options indicating how requirements of the enterprise viewpoint will be met;
- Section 5 presents the Information viewpoint which specifies the rules for defining information models and presents options for their definition;
- Section 6 presents the computational viewpoint referred to as Service Viewpoint;
- In Section 7, the engineering viewpoint is concerned with interaction and distribution of services; and,
- Section 8 discusses platform options as technological basis for the implementation.

Section 2 introduces standards. Appendix D and Appendix E provide abbreviations and terminology used throughout the document.

The relation between the viewpoints in the different standards and in the LifeWatch Reference Model is explained in Table 1 below.

Table 1: Mapping of the RM-ODP viewpoints to LifeWatch

Viewpoint Name	Definition according to ISO/IEC 10746	Definition according to the OpenGIS Reference Model	Definition according to LifeWatch	Dealt with in
Enterprise	Concerned with the purpose, scope and policies governing the activities of the specified system within the organization of which it is a part	Focuses on the purpose, scope and policies for that system	Reflects the planning objectives, strategies and policy issues and reflects the biodiversity research viewpoint of the Infrastructure	Section 3
Information	Concerned with the kinds of information handled by the system and constraints on the use and interpretation of that information	Focuses on the semantics of information and information processing	Specifies the modelling approach and a core ontology to all categories of information of the LifeWatch Architecture	Section 5
Computational	Concerned with the functional decomposition of the system into a set of objects that interfaces enabling system distribu-	Captures component and interface details without regard to distribution	Referred to as “Service Viewpoint”. Specifies the LifeWatch Interface and Service Types that aim at improving the syntactic	Section 6

	tion		and semantic interoperability within the LifeWatch Architecture	
Engineering	Concerned with the infrastructure required to support system distribution	Focuses on the mechanisms and functions required to support distributed interaction between objects in the system	Concerned with aspects related to Reference Architecture: characteristics and principles for (i) services descriptions, (ii) service networks, and (iii) managing service networks.	Section 7
Technology	Concerned with the choice of technology to support system distribution	Focuses on the choice of technology	Specifies the technological options for platforms, their characteristics, and its operational issues.	Section 8

It should be noted that the Engineering Viewpoint encompasses that of the OpenGIS Service Architecture (ISO 19119) and in the ORCHESTRA Reference Model providing a more holistic view in that the characteristics and principles for (i) services descriptions, (ii) service networks, and (iii) managing service networks are considered. The presentation is based on the OASIS Reference Architecture for Service Oriented Architecture, Version 1.0.10

[10] http://docs.oasis-open.org/soa-rm/soa-ra/v1.0/soa-ra.pdf

Blank Page

2 Standards, terms and abbreviations

2.1 Standards

LifeWatch relies on and conforms to published standards whenever feasible. Several standards and best practices related to standards provide a guideline for the LifeWatch ICT conceptualisation and should be adopted whenever feasible and appropriate. Standardisation organisations of particular interest are:

- Biodiversity Information Standards (TDWG)11;
- Open Geospatial Consortium (OGC)12;
- Organization for the Advancement of Structured Information Standards (OASIS)13;
- Open Grid Forum (OGF)14; and,
- World Wide Web Consortium (W3C)15.

Therefore, this document incorporates by dated and undated reference, provisions from other publications. These references are cited at the appropriate places in the text and the publications are listed hereafter. For dated references, subsequent amendments to or revisions of any of these publications apply to this document only when incorporated in it by amendment or revision. For undated references, the latest edition of the publication referred to applies.

2.2 Standards and best practices related to Open Distributed Computing

ISO/IEC 10746-1:1998 (E). Information technology - Open Distributed Processing - Reference model

ISO/IEC 10746-2:1996 (E). Information technology - Open Distributed Processing – Foundations

ISO/IEC TR 14252:1996. Information technology - Guide to the POSIX Open System Environment

2.3 Standards and best practices related to general services architectures

OASIS-SOA-RM16
Reference Model for Service Oriented Architecture 1.0
Committee Specification 1, 2 August 2006

OASIS-SOA-RA17
Reference Architecture for Service Oriented Architecture Version 1.0
Public Review Draft 1
23 April 2008

The OpenGIS Abstract Specification, Topic 12: OpenGIS Service Architecture18

---

11 Formerly the IUBS Taxonomic Databases Working Group, http://www.tdwg.org/
12 http://www.opengeospatial.org/standards
13 http://www.oasis-open.org/
14 http://www.ogf.org/gf/docs/?final
15 http://www.w3.org/
16 http://www.oasis-open.org/committees/download.php/19679/soa-rm-cs.pdf
17 http://docs.oasis-open.org/soa-rm/soa-ra/v1.0/soa-ra-pr-01.pdf
18 http://portal.opengeospatial.org/files/?artifact_id=1221

Standards and best practices of the World Wide Web Consortium (http://www.w3.org/) related to web services.

2.4 Standards and best practices related to ORCHESTRA19

OGC 07-097 "Reference Model for the ORCHESTRA Architecture (RM-OA) V2 (Rev 2.1)", version 2 (Rev 2.1)

"Reference Model for the ORCHESTRA Architecture (RM-OA Version 2), Annex A1 "Requirements for the ORCHESTRA Architecture and ORCHESTRA Service Networks"

"Reference Model for the ORCHESTRA Architecture (RM-OA Version 2), Annex A3 "Conceptual Meta-Information Model"

"Reference Model for the ORCHESTRA Architecture (RM-OA Version 2), Annex B1 "Rules for ORCHESTRA Application Schemas for Meta-Information"

ORCHESTRA Document "Specification of the ORCHESTRA Web Services Platform", Rev. 1.1.1

ORCHESTRA Service Specifications, e.g. "Specification of the Catalogue Service", Rev. 2.2.1

2.5 Standards and best practices related to semantics

World Wide Web Consortium (http://www.w3.org/2001/sw/BestPractices/)

Semantic Annotations for WSDL and XML Schema, W3C Recommendation 28 August 200720

RDF draft standards21

Editor's Note: *To Be Completed.*

2.6 Standards and best practices related to geographic / spatial information

ISO 19101:2004. Geographic information -- Reference model

ISO/TS 19103:2005. Geographic information -- Conceptual schema language

ISO 19107:2004. Geographic information -- Spatial schema

ISO 19108:2004. Geographic information -- Temporal schema

ISO/FDIS 19109:2005. Geographic information -- Rules for application schema

ISO 19111:2003. Geographic information -- Spatial referencing by coordinates

ISO 19112:2003. Geographic information -- Spatial referencing by geographic identifiers

ISO 19115:2003. Geographic Information – Metadata

---

19 http://www.eu-orchestra.org
20 http://www.w3.org/TR/sawsdl/
21 http://www.w3.org/RDF/

ISO 19115-2:2009. Geographic information -- Metadata -- Part 2: Extensions for imagery and gridded data

ISO 19119:2005. Geographic Information -- Services22

ISO 19123:2005. Geographic Information -- Schema for coverage geometry and functions

ISO 19125-1:2004. Geographic Information -- Simple feature access -- Part 1: Common architecture

ISO 19136:2007. Geographic Information -- Geography Markup Language (GML)

2.7 Standards and best practices related to workflows

Workflow Management Coalition (WfMC)23 is a consortium, formed to define standards for the interoperability of workflow management systems

Editor's Note: To Be Completed.

2.8 Standards and best practices related to Grid services

Editor's Note: To Be Completed.

2.9 Legislation of the European Union

Directive 2007/2/EC of the European Parliament and of the Council of 14 March 2007 establishing an Infrastructure for Spatial Information in the European Community (INSPIRE)

Commission Regulation 1205/2008 of 3rd December 2008 implementing Directive 2007/2/EC of the European Parliament and of the Council as regards metadata

COMMISSION REGULATION (EC) No 976/2009 of 19 October 2009 implementing Directive 2007/2/EC of the European Parliament and of the Council as regards the network services COMMISSION REGULATION (EC) No 976/2009 of 19 October 2009

2.10 Terms and definitions

Appendix E contains a list of terms and definitions used throughout the present document.

2.11 Abbreviations

Appendix D contains a list of acronyms and abbreviations used throughout the present document.

---

22 Also published as: "The OpenGIS Abstract Specification - Topic 12: OpenGIS Service Architecture" http://www.opengis.org/docs/02-112.pdf

23 http://www.wfmc.org/

Blank Page

3 Enterprise viewpoint

The Enterprise Viewpoint articulates the requirements that biodiversity research imposes on the ICT infrastructure, inherent requirements for a flexible ICT infrastructure, and requirements that arise from an assessment of the state of the art.

3.1 Stakeholders and generality of LifeWatch infrastructure

Stakeholders in LifeWatch include the following categories:

- Data providers
- Research users
- Policy users
- Media users
- Commercial users
- Public users
- Users in education and teaching
- Ecosystem and environmental resource managers
- Conservation managers.

The primary user groups will be biologists and biodiversity professionals coming from the scientific research community. They will benefit from access to digital data sources and new digital services. They will carry out computational experiments by composing electronic workflows from the operations that are provided by the services.

Although the LifeWatch infrastructure is designed primarily to meet the needs of this primary user group, it is anticipated that the infrastructure will be capable of supporting a much wider range of uses for which supporting biodiversity science might be required. Such uses include, for example:

- Environmental management and control;
- Conservation management; and,
- Support to other European research infrastructures and networks with interests overlapping the biodiversity domain (e.g., Aureoa Borealis, ANAEE, EvolTree).

The new infrastructure will, therefore, offer facilities for biodiversity information and analytical capabilities for specific markets and user group demands, and will develop and offer expert services and products that in return will also create new demands.

Computer scientists and ICT professionals will develop the digital services and tools in co-operation with biologists and biodiversity professionals. The communication between stakeholders and LifeWatch product management / development will be mediated through the LifeWatch Service Centre (e.g., through the User Platform and the Data Providers Platform).

Research networks and research sites may act as data providers and/or service providers to LifeWatch. They may operate LifeWatch core functions on behalf of LifeWatch.

3.2 Planning objectives

When formulating the ICT Construction Plan we keep in mind some key planning objectives to be met. The plan will seek to ensure that we:

- Construct a technical infrastructure that is 'fit-for-purpose'. By that, we mean an infrastructure that is flexible, adaptable, robust, resilient, scalable, and maintainable; and one that meets the needs of the stakeholders;
- Ensure that the infrastructure offers a set of "capabilities" to users that will ensure early engagement with and attract the widest community of users from our stakeholder groups;
- Consider the scientific usage cases that the infrastructure has to meet;
- Consider the operational and technical constraints under which the infrastructure must be built and under which it must function;
- Build an infrastructure that is "open" and based, as appropriate, on widely available industry standards. It will not be proprietary;
- Use existing technological solutions where appropriate and to the greatest extent possible to ensure rapid implementation and deployment (i.e., no re-inventing wheels); but also give room to a parallel experimental approaches in selected areas where research is promising innovations for future applications;
- Take the inherent heterogeneity of the Europe-wide IT landscape supporting research in its stride;
- Establish the infrastructure and become operational at the earliest opportunity;
- Adopt a staged approach to construction and deployment to accommodate all of the long-term desired functionality, recognising that not everything can be available on 'day 1'.
- Take advantage of and integrate "legacy resources", that is resources provided by institutions or networks concerned with biodiversity research.

These planning objectives are essential to create the right balance between achieving the widest pan-European and pan-community support for and use of LifeWatch whilst ensuring at the same time that we can build and deploy the capability in a controlled and manageable manner.

3.3 Strategic approach

When considering how to meet the above objectives we naturally ask a number of questions:

- What capabilities and functions are needed to support the scientific themes, aims, and objectives of LifeWatch?
- What constitutes attractive LifeWatch 'services' that will be of use to the widest community of biodiversity researchers?
- In what order should we deploy and launch services? How should we package groups of services and functions?
- Which capabilities have an ICT component? Which capabilities have a human element to their provision? Which have both?
- What are the semantic issues?
- What are the functions that are common across various biodiversity applications and usage cases?
- What set of ICT capabilities is required to support each such package?
- What are the detailed and common IT mechanisms needed to provide each ICT capability?

These questions address different aspects of the LifeWatch provision. Some questions (near the beginning of the list) are questions for biologists and biodiversity professionals to address. Some questions (near the end of the list) are for computer scientists and ICT professionals to address. The questions themselves and, indeed, the answers to the questions, have different terminologies, concepts and meanings associated with them that are familiar when seen from one perspective but completely alien from another (although many of the same words may be used!). This is inevitable because of the interdisciplinary nature of the preparation and planning work. However, the answers to the questions have to "meet in the middle" so that the right ICT capabilities can be provisioned in order to support the needs of European biodiversity research. The formalism shown in Figure 3 below helps us to address this problem and provides a basis around which to organise plans.

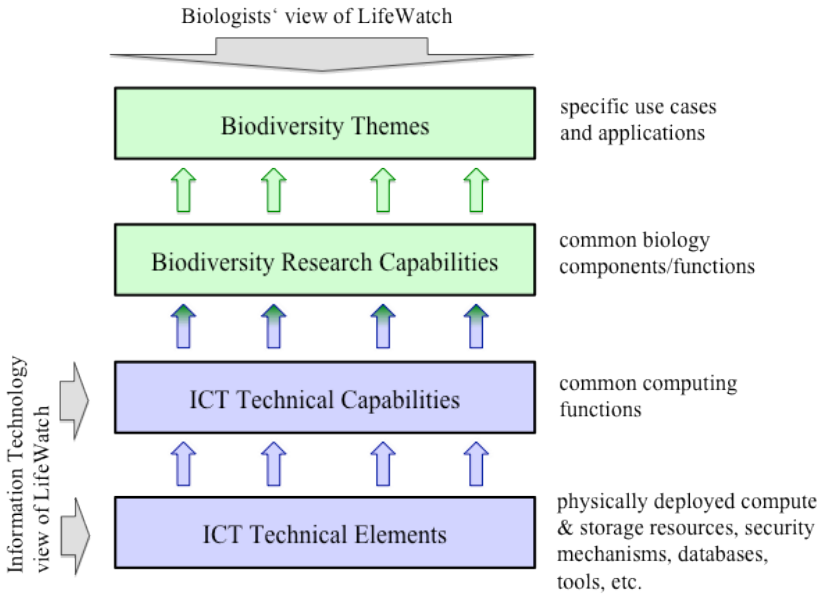


Figure 3: The 4 perspectives of the LifeWatch infrastructure

Starting from the top of in Figure 3, the perspective of Biodiversity Themes allows us to organise thinking according to the scientific themes or research areas of LifeWatch (as described in Section 4 'Scientific Case' of the LifeWatch Concept document, May 2006).

The Biodiversity Themes and some typical showcases will help to identify the Biodiversity Research Capabilities needed. These consist of a set of common biological/biodiversity research components, functions, methods, and abilities from which particular applications and use cases can be constructed. By deriving and deploying a comprehensive set of Biodiversity Research Capabilities LifeWatch enables scientists to compose these capabilities to carry out tasks (use cases) they know today; but at the same time the flexibility will be provided to both add new capabilities and to compose capabilities in ways that cannot currently be foreseen.

The ICT Technical Capabilities are the uniform information and communications technology framework underpinning the Biodiversity Research Capabilities offered to users. Organising thinking and planning in this perspective allows us to derive the common computing and data management functions necessary to deliver the Biodiversity Research Capabilities.

Finally, the ICT Technical Elements are the physical technical components, such as computers, storage resources, communications protocols, security mechanisms, databases, ontologies, tools, etc., that need to be deployed in order to provide the ICT Technical Capabilities of the infrastructure. The precise products and solutions to be deployed in this perspective are technology dependent and can be the subject of specific procurements and projects initiated during the construction phase. Isolating this perspective as a separate view from the ICT Technical Capabilities perspective allows the LifeWatch programme to cope with and adapt to the rapid rate of technological change characteristic of the ICT sector

3.4 Biodiversity research themes

The scientific rationale for LifeWatch24 includes the following themes of research in biodiversity:

[24] See project document: "LifeWatch: From Conception to Realisation, Background information", version February 2007.

Discovery of biodiversity

Species discovery and ecosystem dynamics (in time and space) need new approaches to information integration and knowledge development. Such development will contribute to provide reliable species identifications and to classify species and ecosystems as per their biological relations, functional properties, and evolutionary or ecological history.

Biodiversity patterns - mapping hot spots

We should be able to compare different hotspots by spatial levels, representative taxa and local dynamics and vulnerabilities. Ecological niche modelling is directed at the prediction of actual distribution of biodiversity, based on estimates of the dimensions of potential ecological niches of the components (species). Species ranges are hypothesised by extrapolations of collection sites or observed localities of presence. Other approaches also take into account the value of genetic components by introducing phylogenetic weighting in biodiversity mapping. Large-scale and reliable information is essential here.

Biodiversity processes - monitoring changes

A large obstacle in biodiversity research is the absence of adequate long-term and comparable time series data about biodiversity changes. Monitoring changes in Europe's biodiversity at an appropriate scale is essential to understand and predict processes. Standards and "best practice" approaches are one part of the solution. Historical data from natural history collections and literature, new data collection, processing, fusion and representation is another.

Systems biology

Comparative data mining in large data sets from the different (genetic, population, species and ecosystem) levels of biodiversity allows us to identify patterns shared between these levels and provides evidence for the mechanisms behind them.

Nature conservation and management

Data and knowledge about the characteristics of spatial distributions, changes at different time scales, and the associated multi-state dynamics, provide a more solid basis for making decisions. Implementation of the decisions and monitoring their effects becomes more reliable.

LifeWatch will promote showcases as examples of the kinds of scientific studies that biodiversity researchers would like to be able to undertake within the context of the above themes.25

3.5 Biodiversity research capabilities

The Biodiversity Themes allow identifying some common functions, referred to as "Biodiversity Research Capabilities". These functions may be shared by several Research Themes or may be specific for some theme. Typical such functions, grouped according to several kinds of functionality, are shown in the list below.

Data discovery and access

- Search for data such as species occurrence data, natural history collection data, DNA sequence data, past climate data, or genome sequence data on specified attribute and/or relation and access these data
- Specify thematic, spatial, or temporal constraints on search
- Manage access rights / show access and licensing rights
- Send data or enable access to data in specified formats

[25] See *LifeWatch Show Cases* (in preparation)

- Evaluate “fitness for purpose” and data quality constraints of data sources

Data processing

- Provide simple tools to explore and map relationships between attributes and pressure/drivers
- Compose species occurrence data into species distribution map
- Obtain the ecological envelope for a species
- Produce a phylogeny from DNA sequence data
- Combine phylogeny, geographic, and temporal information

Modelling

- Discover models or algorithms for spatial or temporal modelling
- Simple online capability to model and map relationships between biodiversity attributes and pressures in relation to forecasts
- Interpolate / extrapolate data in a region based on existing data points
- Use predictive models based on physical habitat data to predict species distribution

Visualization

- Display and interact with species distribution map on basis of GIS map layers
- Evaluate map layers according to set valuing criteria
- Display graphs of concepts (and their relations)

Support for user and user groups

- Provide list of acknowledgements for all data used so that they can be properly acknowledged and cited
- User feedback mechanisms to provide assessments of quality of data and services
- Record the automated and manual processes undertaken to reach a decision / conclusion
- Collaborate with a large group of diverse stakeholders over months and years
- Provide audit trail or workflow documentation

The capabilities in the above list span several different scientific subjects (taxonomy, ecology, genomics, climatology, geography, for example). This is because biodiversity research is a multi-disciplinary domain. Each capability may be functionally atomic in character (e.g. "Access past climate data", "Create GIS map layers") or it may encapsulate multiple atomic functions to describe an entire research task (e.g. "Use predictive models based on physical habitat data to predict species distribution"). In addition, in either case capabilities incorporate a variety of common computing and data management functions. They are a proxy for the underlying ICT capabilities that makes them possible. The relation between biodiversity themes, research capabilities, showcases, and scientific subjects is schematically sketched in Figure 4.

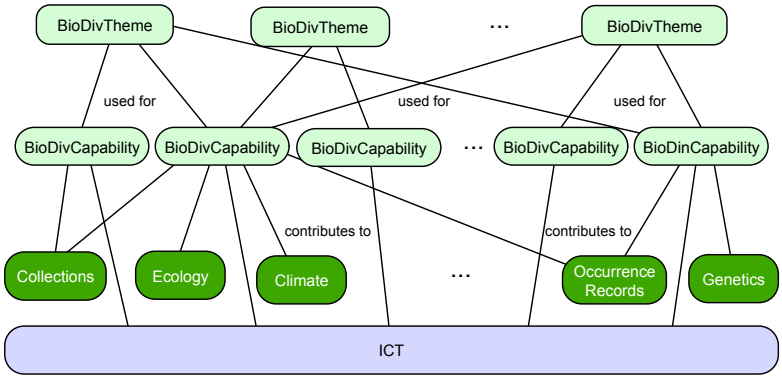


Figure 4: Schematically relating the components of biodiversity research

3.6 ICT capabilities

The ICT capabilities subsume common computing and data management functions necessary to deliver the biodiversity research capabilities. The list of required ICT capabilities is grouped according to different technical aspects and is by no means exhaustive.

Functional requirements

The functional requirements concern the types of operations that the users need in order to find, access, process and view data. They include:

- Searching and browsing mechanisms for distributed data and services.
- Uniform identity framework for data and services
- Access to existing data and services, distributed among multiple organisations. Data and service providers continue to manage their data (and services) independently as now, including control of the creation and modification of data/services. However, data can be accessed by authorised users located anywhere through a generic mechanism defined by LifeWatch.
- Mechanisms for source data preservation, i.e., access to past versions of data sets that have been used to produce secondary information.
- Capture data from users and lightweight devices, including field sensors and networks providing continuous streams of new data, and portable computing devices, often with intermittent connectivity.
- Mechanisms for data analysis as well as mapping and modelling tools, using standard ways to manipulate and view data.
- Mechanisms for data fusion, integrating different sources (such as sensor data, biodiversity parameters, geographic data, primary data, workflow execution), to allow fast retrieval at different levels of detail e.g. for analysis and visualisation.
- Support the understanding of results by the user, by providing tools and mechanisms to enhance knowledge extraction from discovery as well as from analysis results

Composition requirements

The types of operations identified in the functional requirements are the elements that users can compose into electronic workflows. The operations require input and produce output in specific forms. Therefore, it may be necessary to translate/mediate between data formats and semantics. The output of a workflow is a new piece of data whose provenance must be documented. This documentation must be at an appropriate level of detail to permit critical review and re-use; and must remain correct even if source data are changed. Finally, the usage of data in a workflow gives information about the use of data and possibly its fitness for the purpose of the workflow. Composition requirements include:

- Mediation mechanisms like a semantic metadata framework should enhance interoperability between resources.
- Support for workflow modelling, allowing the arbitrary composition of services and resources, providing documentation and enactment of data processing steps. Workflows can be stored in libraries, allowing them to be shared in the same controlled manner as data.
- Creation and management of provenance information, documenting the use of data and workflows from collection to publication. Traditional provenance allows tracking of the source of derived data and providing reproducibility of results. We include the requirement for “reverse provenance”, showing where data has been used, to support licensing and to guide future data collection. Support must also be provided for links to external resources, as well as a standard format to be applied in external resources to link to LifeWatch data.
- Mechanisms for managing the contradiction between, on the one hand,
    - The requirement to allow data and tools to be updated independently by their providers with, on the other hand,

- The provenance requirement to be able to identify and recall a specific version of both the data and the tools used in an analysis in order to examine and reproduce the analysis.

Collaboration requirements

LifeWatch shall support collaboration between researchers. Not only indirectly, by accessing remote data and sources, but also directly: by forming projects, communicating in groups, sharing and discussing results, sharing knowledge in restricted groups, etc. Collaboration requirements include:

- Support for building and managing virtual organisations by allowing the definition of groups and roles and their authorisation for sharing resources;
- The need for collaborative community support tools to allow groups of people to work on a shared objective. Support should be provided for collaborative tools that have been found to be useful in existing projects: mailing lists, wikis, weblogs, publish/subscribe web feeds, and social networking elements. Virtual laboratories, ("common exploratory environments") will bring together communities, collaborative tools, data and process workflows to provide, through component-based user interfaces (portals, mash-ups, etc.) a platform for shared scientific research;
- Allowing users to add annotations to existing data and services, which may contribute to the quality assessment and feedback processes;
- Control mechanisms for access to data and services, together with monitoring services for the support of service level agreements and, potentially, including charging mechanisms.

The ICT Capabilities are provided in terms of services and workflows. Services are the execution units from the perspective of the reference model and the basic building blocks of a service-oriented architecture. Services can be composed ("chained") to become workflows. Workflows may be obtained by a sequence of user actions each invoking a service or, at the other extreme, may be specified as a formally specified composite service.

In LifeWatch, workflows are the means to relate the biodiversity perspective and the ICT perspective. Workflows implement Biodiversity Research Capabilities. This is illustrated in Figure 5.

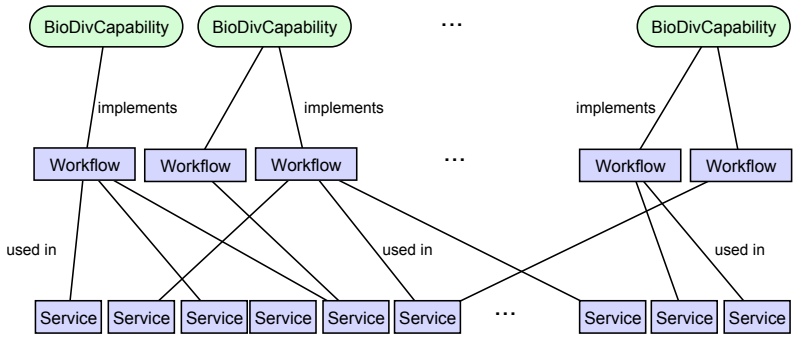


Figure 5: Relating biodiversity and ICT perspectives

Sometimes a single service will provide a complete Biodiversity Research Capability. In other cases, several services may need to be composed / chained in sequence to provide a more complex Biodiversity Research Capability. Biodiversity Research Capabilities, being themselves services, can also be composed / chained into workflows.

From the perspective of the biodiversity researcher, workflows represent the equivalent of experiments as, for instance, documented in a laboratory journal. Workflows can be shared. They can be reused to reproduce and thus check results, as they are an objective basis for the discussion of methods and methodologies, thus supporting a new level of scientific practice.

3.7 Conformance to standards

The Infrastructure for Spatial Information in the European Community (INSPIRE) is an initiative of the European Commission to establish a European spatial information infrastructure as a basis for services providing access to spatial information held by public authorities. The INSPIRE Implementation Rules detail the INSPIRE legislative requirements for data and metadata so that these can be implemented consistently across Europe.

The LifeWatch information models shall aim to conform with the INSPIRE Implementation Rules so that LifeWatch can access all INSPIRE-conformant data and services and can itself provide INSPIRE-conformant data and services.

GMES and GEOSS will be important sources of earth observation data. GEOSS, in particular, has a focus on biodiversity data acquisition with the long-term aim of integrating a distributed biodiversity observation network with sectorial, crisis, health, and policy systems. It can be expected that all these data will be made available according to the INSPIRE Implementation Rules, hence will be accessible for LifeWatch.

The Open Geospatial Consortium (OGC) is an influential consortium for geospatial standards, and many spatial data are provided through OGC services. OGC has agreed to develop INSPIRE-conformant service specifications. OGC accepted the ORCHESTRA Reference Framework that will serve as a basis for LifeWatch as a 'best practice'.

The Open Grid Forum (OGF) is an important source of standards and specifications for applied distributed computing fundamentals (i.e., Grid computing technologies, e-Infrastructure, etc.) that will form the foundation of the LifeWatch infrastructure.

Formerly known as the "Taxonomic Data Working Group", Biodiversity Information Standards (TDWG) is an international not-for-profit group that develops standards and protocols for sharing biodiversity data. LifeWatch shall try to follow its recommendations where appropriate. For biodiversity data domains not yet covered by TDWG, LifeWatch will be well positioned to contribute to the TDWG process in order to achieve internationally agreed specifications.

Geography Mark-up Language (GML) is the data exchange format of OGC and INSPIRE. LifeWatch should try to adopt GML for processed spatial data. If possible, the INSPIRE application schemas in GML shall be used.

Ecological Meta-data Language (EML) is the meta-data specification for the documentation and exchange of information about ecological data. LifeWatch shall adopt EML as a meta-data specification.

It is likely that LifeWatch will need to develop additional meta-information definitions, because of its wide scope. GBIF, for example, only supports the four first levels of the recognised 6 levels of EML of increasing completeness: identification, discovery, evaluation, access, integration, and semantic use. LifeWatch will support all 6 levels.

3.8 Considerations on biodiversity informatics

Editor's Note: *The contents of the following sub-sections of 3.8 should be considered as "preliminary" in the current version (v0.4) of the present document.*

The complementary LifeWatch report on the state-of-the-art in biodiversity informatics provides an assessment of existing research infrastructures, networks, data exchange formats and models, services, workflow, and grid environments. It makes recommendations concerning the interaction of these with the LifeWatch infrastructure. The need for interaction between LifeWatch and the present world of biodi-

versity informatics imposes additional requirements on the LifeWatch infrastructure. Some of these, in relation to some general principles regarding handling of data within LifeWatch, are considered in the sub-sections below.

- Transformation of existing standards into workable formats
- Use of EML
- Ability to output and forward data in original formats
- Need to support standardisation efforts in order to cover the entire LifeWatch domain.

All are in need of further discussion and the details following from these principles will be worked out subsequently.

3.8.1 The Biodiversity Data Domain

3.8.1.1 Biodiversity Information Networks and Data Providers

There are different kinds of biodiversity information networks (communities) and data providers that can be divided into sub domains, largely following the established research communities that LifeWatch will unite: terrestrial ecology, marine ecology, taxonomy, molecular biology and genomics.

Long-Term Ecological Research (LTER) networks and marine networks, comprised of multiple data providers, are important sources of terrestrial and marine ecological data respectively. Natural history museum and other taxonomic collections are important sources of species data, including observations information about the occurrence of such species. Databanks for molecular biology and genomics are increasingly important data sources. Each of these categories of data provider is described in more detail in the following sub-sections.

Data may also come from any user who uses LifeWatch tools to analyse their own data, if this user agrees to share their data in (or through) LifeWatch with others i.e., any user can also be a data provider.

LifeWatch has to make provisions to interact with data providers and data provider communities (networks) of all categories. Our approach is to treat networks as intermediaries between data providers and LifeWatch and to use them as communications channels ('go-betweens') to collect and consolidate data provider requirements and concerns, and to disseminate LifeWatch information to data providers.

3.8.1.2 Terrestrial ecological information

Editor's Note: To be further described according to the topics below.

a) State of the art (knowledge of the entire domain)

International LTER nodes are important sources of ecological data. National biodiversity networks fulfil a similar function.

GEOSS will become an important source of data.

Continuous data streams coming from autonomous terrestrial environmental sensor networks will become increasingly common.

b) Characterisation of overlaps with other domains

c) Description of subdivisions, each of those in turn with a brief description of the sub domain itself, a description of existing efforts to provide standards/data access, etc., and a brief indication of LifeWatch's position

3.8.1.3 Marine ecological information

Editor's Note: To be further described according to the topics below.

a) State of the art (knowledge of the entire domain)

International LTER nodes are important sources of ecological data. National biodiversity networks fulfil a similar function.

GEOSS will become an important source of data.

Continuous data streams coming from autonomous terrestrial environmental sensor networks will become increasingly common.

b) Characterisation of overlaps with other domains

c) Description of subdivisions, each of those in turn with a brief description of the sub domain itself, a description of existing efforts to provide standards/data access, etc., and a brief indication of LifeWatch's position

3.8.1.4 Taxonomic information sources

This domain has been discussed in its entirety for more than two decades now, largely reflected in the activities of the Taxonomic Databases Working Group (Now: Biodiversity Information Standards, TDWG). A number of comprehensive reference models and data standards cover the sub domain, so that information structures and item-level semantics can be considered well understood. The most recent comprehensive coverage is provided by one of the LifeWatch member networks: the Common Data Model of the EDIT Platform for Cybertaxonomy, which incorporates the existing standards and information models as well as the experiences gained from their usage in practice.

The sub domain overlaps with geographic information in species distribution and species occurrence data, with ecological data (e.g. traits) in the descriptive part, with bibliographic data in taxon citations and general references, with molecular data in areas such as DNA bar coding and phylogenetic reconstruction. We should further explore the following subdivisions:

a) Taxonomic backbone: taxon names and synonyms

The taxonomic backbone consisting of the names of organisms and their classification into taxonomic groups provides an essential indexing mechanism for most aspects of biodiversity. On the one hand, the formal rules of nomenclature provide an excellent normative construct which has led to clear data structures and internationally accepted rules for the application of names, on the other hand have more than 250 years of describing and formal naming of organisms added a certain amount of complexity to the issue. In addition, as everywhere when dealing with the living world, the named classes represent scientific hypotheses rather than real-world objects, so that the classification changes over time.

A number of efforts are under way to provide taxonomic backbone information systems. On the global level, the Catalogue of Life26 is a major portal for taxonomic information, which provides a consolidated taxonomic view on about half of the known species on earth. For taxon names, several global "nomenclators" gather data on the existing names for algae, fungi, plants, and animals, usually without providing a judgement on the status of the name (accepted or synonym). On the European level, three projects have gathered the information on the species in Europe: Fauna Europaea for animals, Euro+Med plant base for vascular plants, and the European Register of Marine Species (ERMS). Recently (2008), the three databases come together under a project initiated by EDIT: the Pan-European Species-inventories Infrastructure, PESI27, which will provide unified access to these databases. In addition, numerous national standard taxonomic lists are in use, which are often important in the context of administrative or legal requirements.

From all this, it becomes clear that LifeWatch should not try to compose its own taxonomy from different sources. Biodiversity research will always need to critically revise the taxonomy of the organisms involved in the specific study. LifeWatch should come to an agreement with the different sources of taxonomic backbone information so as to be able to offer their usage in the LifeWatch framework, e.g. in order to disambiguate the species names in queries. In addition, LifeWatch should work closely with all initiatives in the domain (including GBIF, the Global Names Architecture, Encyclopedia of Life, Species 2000/ITIS Catalogue of Life, Pan-European Species directories Infrastructure (PESI), and national authorities) to provide the taxonomic capabilities needed for its purposes.

b) Specimens and species occurrence observations

The taxonomic work process is largely based on – or at least related to - specimens and their collection records. Specimens provide a falsifiable base for the hypothesis that is articulated in the description of a species. Both specimen data and collection metadata ("where can I perhaps find material for my study") are thus essential for taxonomic research. The data domain is very broad, ranging from data on the collection event (e.g. local usage, common name, and partial description of individuals, behavioural observations of behaviour, ecological notes) to chemical or molecular data gained from the specimens. In addition to specimen-documented occurrence records, species observations that should be counted as belonging to the taxonomic domain include the data recorded for faunistic and floristic mapping projects (e.g. Atlas Flora Europaea), which have been coordinated by (professional or "amateur") taxonomists

GBIF28 is the global effort to provide access to specimen and species occurrence data (called "primary biodiversity data" in GBIF terminology) through "a global distributed network of interoperable databases". It uses or has developed protocols and mechanisms for data standards and data sharing in cooperation with other networks such as BioCASE and IOBIS (EurOBIS). Since the latter now also offer data via GBIF, our recommendation is first to analyse these mechanisms for (re-) usability in LifeWatch and then to extend them as necessary to meet the specific needs of other networks and individual data providers as necessary in the LifeWatch context. We envisage, for example, that LifeWatch will need more meta-information because of its greater scope (see note in 3.7 above).

c) Taxonomic output: Descriptive data and identification keys

Perhaps the most visible result of taxonomic work processes is the publications of descriptions of taxa (i.e. classes of organisms) in taxonomic monographs, faunas and floras. The terminology used to describe taxa (and consequently also used in the identification keys that allow others to determine the class of organism an individual belongs to) is very extensive, multilingual, has changed over time, and strongly depends on the larger group of organisms. However, taxonomic publications are usually highly structured, and the description itself uses a formalised language that lends itself to data processing.

---

26 http://www.catalogueoflife.org/
27 http://www.eu-nomen.eu/pesi/
28 http://www.gbif.org/

Numerous data sources exist that provide or are based on structured descriptive information, mostly using DELTA (Descriptive Language for Taxonomy), an early TDWG standard. Over the past years, the SDD (Structured Descriptive Data) subgroup of TDWG has developed a new standard (XML based) that is increasingly supported by descriptive data processing tools in use. However, provision of descriptive data is still largely an effort made by individual taxonomists. The Key2Nature project joins several European initiatives (terrestrial and marine) dealing with descriptive data. It collects existing tools and identification keys with respect to European taxa and makes them available to a broad audience.

LifeWatch will need to provide the means to identify species for various purposes. Collaboration with the Key2Nature, EDIT and other projects to identify the medium term infrastructural requirements to support the provision of species identification services and their incorporation into the LifeWatch framework is envisioned.

3.8.1.5 Molecular biology and genomics data sources

Editor's Note: *To be described in a later version of the present document.*

3.8.2 Auxiliary data domains

3.8.2.1 Geographic data

Editor's Note: *To be described in a later version of the present document.*

3.8.2.2 Abiotic data

e.g. Climate, Soil, & Water data

Editor's Note: *To be described in a later version of the present document.*

3.8.2.3 Bibliographical data

Editor's Note: *To be described in a later version of the present document.*

3.8.2.4 Legal data

Editor's Note: *To be described in a later version of the present document.*

3.8.3 Biodiversity data formats

ABCD (Access to Biological Collections Data) and DwC (Darwin Core) are presently the most popular data exchange formats for species occurrence data. Both are recommended by TDWG. These formats should be used to obtain occurrence data from LifeWatch data providers. Moreover, LifeWatch should be able to forward the data as obtained.

On the other hand, ABCD and DwC documents may be very large. Therefore, Ecology Markup Language (EML) descriptions should be developed as an alternative for ABCD and DwC, so that the data can be discovered via these descriptions and then loaded as simple tables (CVS data). This has been proposed by GBIF. EML is the metadata format for ecological data in the USA and some other non-European LTER nodes. LifeWatch should use EML as the exchange format for metadata on datasets.

Taxonomic Concept transfer Schema (TCS) is a standard for the exchange of taxonomic data between different taxonomic data models i.e., different data models used by different providers of taxonomic data. The taxonomic providers of LifeWatch should be encouraged (where they have not already done so) to adopt TCS as the basis for responding to taxonomic queries29.

Like OBIS, LifeWatch should not only access and process occurrence data, but also animal tracks as lines, time series data, abundance, and absence data. LifeWatch must provide and support appropriate data formats for such kinds of data.

One focus of LifeWatch is on data integrating and processing where data integration typically relies on data that can be located in space and time. Hence, LifeWatch should support a uniform data form for spatio-temporal data.

3.8.4 Data transmission / protocols

TDWG recommends TAPIR for ABCD and DwC. TAPIR is a Web Service protocol to perform queries across distributed databases of varied physical and logical structure. It was originally designed for usage by federated networks. TAPIR was developed for use with biodiversity and natural science collection data but is a generic tool applicable to other domains. Because of its flexibility, TAPIR should be considered as a basis for the protocol development of LifeWatch. It will be necessary to provide support and a migration strategy for those data providers not presently supporting TAPIR.

GBIF itself provides occurrence data in DwC or KML through a REST service. This might be an alternative approach to be investigated in LifeWatch.

As some occurrence datasets are very large, GBIF proposes to support simple delimited files that represent the full dataset and are produced on the provider side using a simple database export and then compressed for transfer. These files would be described in EML. LifeWatch may adopt this strategy, but the more general question should be raised of how to transmit large and/or complex data sets or data streams as may occur for sensor applications.

3.8.5 Globally unique identifiers

Generally, any data that is generated by LifeWatch, or that is retrieved into the LifeWatch infrastructure and subsequently processed, should be identifiable by a globally unique identifier (GUID) as a foundation for provenance and citation.

TDWG promotes Life Science Identifiers (LSIDs) as the way to uniquely name and locate (identify) pieces of biodiversity information on the web. Recently there has been some discussion whether HTTP URIs can perform a similar naming task.

LifeWatch should investigate the issue further.

3.8.6 Semantic aspects

Data exchange formats define the syntax of a representation. In addition, a semantic model is needed in order properly to exploit the relations between concepts expressed in different datasets and tools. With common vocabularies, expressed through ontologies, the inputs and outputs of services can be described in such a way that chainable services can be detected and even connected automatically. This can relieve

[29] It is likely that TCS will need to be extended, based on the experience of developing the Common Data Model within EDIT.

the user of a significant burden in attempting to resolve compatibility issues between services. Furthermore, semantic annotations can support discovery, mediation, and integration of (meta) data, services, workflows, etc..

TDWG is presently developing ontologies and LSID vocabularies for biodiversity information that should be used for that purpose.

OWL/RDF presently seems to be the favourite formalism for semantic conceptualisations, being used by TDWG, ALTERNet (SERONTO), and other ontologies. LifeWatch shall adopt these formalisms.

Vocabularies for the various sub-domains of biodiversity are most likely to overlap and there is also potential for conflict arising out of the use of the same term to represent different concepts in different sub-domains. Taxonomy is a clear example of where such problems occur. Mechanisms for the resolution of such conflicts will be required.

3.8.7 Intellectual property rights

ABCD supports extensive metadata on intellectual property rights (IPR) and other rights, thus ensuring that data providers can make their claims as to copyright, proper accreditation, and utilisation of their data. LifeWatch should extend IPR information to all kinds of data, requiring it for primary data as well as generating it for derived data. Such information may be represented as meta-information related to a globally unique identifier (GUID) for the data.

3.8.8 Provenance

Provenance, or lineage as it is sometimes called, comprises information about the derivation from particular sources to the present state of an entity. Provenance distinguishes between the source (or derivation) of an object and the recording of the derivation. In a nutshell, one might say that provenance records

- What is the sequence of events that led to the present state of an entity?
- When did the events take place?
- Where did the events take place / did information come from?
- How was an entity was obtained? What are the actions involved?
- Who are the agents involved?
- Why did the events take place?
- Which is the set of devices involved?

This induces clear requirements related to provenance meta-data. It should be possible to:

- Trace the origin of data and workflows used/cited within LifeWatch (backward direction towards point of origin); and,
- Track the use and re-use of data and workflows provided into the LifeWatch infrastructure (forward direction towards point of academic publication).
- The first mechanism is needed in order to reproduce results claimed by publication reliably. The second mechanism is needed in order to keep track of who is using a particular provider's intellectual property.

Although significant research has already been accomplished, the principles of provenance in e-Science are a relatively new topic of study. See, for example, the work of the UK's e-Science Institute theme on this topic30. Therefore, this is presently an area for further investigation within LifeWatch.

3.9 Community building

The success of LifeWatch depends on the participation of its user and contributor communities. A user basis will be attracted and sustained by the quality and stability of the resources - services, portals, tools, data, etc. - offered by the LifeWatch. Achieving a sustainable momentum for contributors is a more difficult task. Data providers will see the benefit of having access to a wider database and, given that sufficient support is given, start and continue to provide data. The situation may be more diffuse when considering contributions from Biodiversity Informatics:

Since it is to be expected that the development of the LifeWatch ICT infrastructure essentially proceeds in terms of many nationally funded projects with LifeWatch being the coordinating factor apart from providing some basic infrastructure construction. The LifeWatch Reference Model aims for being a unifying framework to coordinate these projects. However, (often competing) construction work in different projects fosters the not-invented-here syndrome familiar in computer science. The experience is that sustainable contributor communities in computer science build around open-source community-based projects. LifeWatch will promote the building of such a community by providing its software and tools on an open-source basis.

[30] http://wiki.esi.ac.uk/Principles_of_Provenance

Blank page

4 Scope and structure of the LifeWatch ICT infrastructure

4.1 Introduction

The technical vision for LifeWatch is a vision of a network of services providing secure access across multiple organisations to biodiversity and related data and to relevant analytical and modelling tools to collaborative groups of researchers. The infrastructure will be implemented using the ideas of "ORCHESTRA Service Networks", Spatial Data Infrastructures, and Grid Computing.

The overall architectural concept can be derived from the requirements listed in the enterprise viewpoint (Section 3). They are reprised in the following requirements:

Technology independence and rigorous use of standards

The use of open standards allows the construction of a large system as an "ecosystem" of loosely coupled independently developed components. The use of open, standard components reduces the risk that a failure to implement one component successfully will prevent the successful deployment of the remaining system and will ensure vendor independence. The sharing of geospatial data should be realised through standard concepts and interfaces developed by the Open Geospatial Consortium (OGC).

Loosely coupled components

The application of the Service Oriented Architecture (SOA) paradigm will assist in overcoming three primary issues in distributed computing: heterogeneity, availability, and scalability. The integration of widely deployed security and sharing mechanisms from the Grid community, and particularly from established European e-Infrastructures, will ensure interoperability with data and tools from outside the biodiversity field. These mechanisms must not only enable the sharing of resources, but also allow resource owners to control access based on identity, licensing, and remuneration. Mediation mechanisms should allow that existing components could be interconnected without changes. An approach for integration of diverse data will be the use of ontologies from the biodiversity communities such as TDWG and LTER and the mapping of common features of these ontologies to some core ontology.

Generic, self-describing infrastructure components

The LifeWatch Infrastructure will provide a set of domain independent services (architectural services), which can be used across different domain applications and organisational contexts. All components should provide a description of their critical characteristics. The specification of meta-information models, will allow the independent association of arbitrary metadata with existing data and services without modification of the existing resources.

Unambiguous identification

All infrastructure components should be attached to an identification mechanism, which should provide an unambiguous authenticated and authorised access to the resources (e.g. data, meta-information and services). Core LifeWatch functionality such as discovery, access, provenance, and personalisation will be based on this property.

Evolutionary development

The architecture (including the interfaces) should be flexible enough to evolve during and after the construction phase. It should be adaptable to changing user and technological requirements. The use of standardised, semantically described service interfaces contributes to ease of maintaining the underlying implementations. Workflows will help to codify the use of tools and methods, as well as to implement and deploy new services by chaining existing ones. The publication of the description of research tasks in a workflow repository will encourage reuse and validation of scientific methods.

Multiple representations

Different users and different application domains of the infrastructure demand for decoupling information from presentation. Even for a single application domain, the presentation can vary according to a specific usage context or user preferences. LifeWatch will support this requirement by separating mechanisms for presentation and user interaction from services for data access and processing. These services can be accessed through a general-purpose portal. They can also be used (and extended if necessary) for building applications to support higher-level and more problem-specific interfaces for domain-specific tasks.

This section introduces the overall architecture of the system and describes mechanisms to achieve the aforementioned architectural requirements. First, the functionality of the infrastructure according to "functional domains" as defined in the LifeWatch Masterplan. Then several aspects are highlighted that contribute to the requirements outlined above. This assists the reading of the subsequent sections dealing with the Information and Services Viewpoints (Section 5 and Section 6),

4.2 Functional domains

4.2.1 Overview

The definition of functional domains for a service network allows a classification of the infrastructure components according to interaction mechanisms. This description appeals to the user perspective. Figure 6 shows the LifeWatch functional domains, which will be further explained in this section

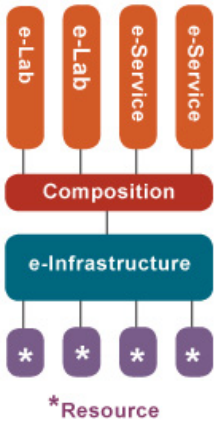


Figure 6: Functional domains for the LifeWatch architecture

In Figure 6 four layers are depicted, which can be each related to the functional domains described by ORCHESTRA, as follows:

- The Resource Layer contains the specific resources, such as data repositories, computational capacity, sensor networks / devices and modelling / analysis tools, that contribute to the LifeWatch system. The primary components at this layer are the data from sites and collections, but additional components include catalogue services (e.g. taxonomic checklists or gazetteers), analysis tools, and processing resources. In the RM-OA, this is called the Source System Domain.
- The Infrastructure layer provides mechanisms for sharing the specific resources as generic services in a distributed environment spread across multiple administrative domains. This includes systems for identifying, accessing, and processing resources located within multiple administrative domains, uniform security and access protocols, and ontologies for semantic integration. In the RM-OA, this layer is split in two: the Integration Domain and the Mediation and Processing Domain.
- The Composition Layer supports, through the use of a semantic metadata framework for unambiguous discovery and provenance recording, the intelligent selection and combination of ser-

vices in order to complete tasks. This includes workflows, semantic metadata for the discovery of components and the storage of additional attributes such as provenance and version information. This is a specialised part of the RM-OA Mediation and Processing Domain.

- The User Layer provides domain-specific presentation environments for the control and monitoring of tasks and tools to support community collaborations. This includes a primary LifeWatch portal incorporating workflow management tools, but also domain-specific portals. In the RM-OA, this is called the User Domain.

4.2.2 Resource layer

The Resource Layer contains a variety of data, processing tools, and instruments. These primary resources are managed by multiple organisations, and in general, LifeWatch cannot dictate their location or configuration. The resources can be mainly grouped into four categories.

4.2.2.1 Biodiversity data

The integration of biodiversity data resources faces similar problems as other fields. In addition, biodiversity data often has geospatial, temporal or taxonomic attributes, which can make discovery of appropriate data more difficult. Such data is exemplified by occurrence data, which record the observation of one or more organisms of a particular species at a particular location and time. As specified in the informational viewpoint, the LifeWatch mechanism for integrating biodiversity data is the definition of information models and the mapping of data resource types to feature types, structured by an application schema. An information model specifies new data structures in terms of simpler data structures or basic data types. Key data structures for the LifeWatch infrastructure are geospatial and temporal features, as defined by the ORCHESTRA Reference Model, and biodiversity data. These concepts will be drawn together through LifeWatch ontologies.

4.2.2.2 Biodiversity tools

The integration of biodiversity tools is also a complex theme, since at the time of writing this document there are a number of heterogeneous and non-standardised tools for research tasks. The integration of a specific tool should be done on the application level for a particular domain. Nevertheless, the infrastructure will provide architectural services to support this development, for instance for the call of external applications or for the definition of transfer protocols to simplify the wrapping of tools as LifeWatch processing services.

4.2.2.3 Biodiversity services

The existing biodiversity networks and projects have already defined and implemented web services to provide access to biodiversity data and methods. The LifeWatch infrastructure will consider existing generic services when defining the LifeWatch interfaces. With the help of schema mapping services and other mediation mechanisms it should be easy to attach such existing services to the infrastructure, e.g. for taxonomic search or validation.

4.2.2.4 Computational equipment

The LifeWatch infrastructure is a distributed system and should provide management mechanisms to allocate computational resources for processing and storing data. Platform neutral interfaces should be provided, inspired by mechanisms from existing GRID technology.

4.2.3 Infrastructure layer

The role of the Infrastructure layer is to impose a layer of syntactical and semantic uniformity on the heterogeneous resources in the Resource Layer, to enable the sharing of resources located at different organisations and to provide the functional capabilities of the Infrastructure.

Functional components in LifeWatch are implemented as services. A Service-Oriented Architecture (SOA) allows the definition of uniform interfaces to implementations providing the same capability. This allows services to be substituted transparently, and so increases the availability of capabilities when some resources are busy or unavailable. The Services model does not require a one-to-one mapping from services to resources, allowing services to be reused.

The Services Viewpoint classifies possible functional components in the terms of services sub-categories and service types. Figure 7 below gives an overview of these sub-categories, thus structuring the Infrastructure layer. The parameters used by the services interfaces are data structures defined by information models, as described in the Information Viewpoint. The access to existing resources from the resource layer is done by mapping them syntactically and semantically to specified information models. A possible mediation mechanism to enhance this process to achieve semantic interoperability is described in the Information Viewpoint (Section 5).

An important aspect of the Infrastructure layer is the scalable management of users and their access to resources (authorization, authentication and accounting). This covers the requirement for a common identity and authentication framework. However, there is also a requirement that the framework be scalable, from the point of views of both user management and resource owners. Collaborators working toward a common goal can be managed as an organisation in their own right. Since the collaborators may come from different physical organisations, we refer to this structure as a Virtual Organisation (VO)31. Management of users is based on their membership of, and role within, a Virtual Organisation. This makes the authorisation tasks of resource owners simpler, since they do not need to be aware of changes in the users' roles, merely of the access permissions for a role within a VO.

4.2.4 Composition layer

The Composition Layer provides the tools for intelligent selection and orchestration of services in order to perform a given task.

Composing multiple services can create new capabilities and these new capabilities can themselves be published as new services. The user of such a composite service is only aware of the service's interface and defined behaviour, not necessarily how it is implemented by lower-level services.

In order to compose a group of services for their own tasks, users should see the connections and dependencies between sub-components of the computation. Workflow composition environments provide a convenient graphical means for the orchestration and enactment of multiple services in sequence.

Workflows are typically structured using a declarative format. Hence, the workflow can itself be stored as data. This makes it possible to share workflows between users. The LifeWatch infrastructure provides a workflow repository for this purpose. Users may modify parts of existing workflows in order to vary their experiments, or to change the set of data to which the workflow is applied. By sharing a workflow, it is possible for other users to verify the results and methods of an analysis, or to re-apply it to different data.

---

[31] Also known as a Temporary Collaborative Network (TCN)

The LifeWatch infrastructure provides an independent repository for semantic metadata. Access to resources through LifeWatch gives access to the metadata stored in the independent repository. The use of a separate repository allows the metadata to be provided without modifications to the original sources. It also allows the metadata to contain information that may not be approved or supported by the original source. This may include, for example, information on the quality, or information created by a separate community of users. Examples include support for alternative taxonomies for a data resource which assumes a particular taxonomy, or where information about a species can be linked with companion species (e.g. parasites, pollinators).

One of the key purposes of the metadata is to store provenance information with details of the data sources. In LifeWatch, the idea is that provenance information will be collected not only for source data, but also for all derived data. Provenance of derived data includes details of workflows and the source data they are applied to. The provenance of metadata is also useful for determining the validity of metadata attributes or for resolving conflicting metadata (which may be due to different scientific opinions).

4.2.5 User layer

The User Layer provides the high-level interfaces for the users of the LifeWatch system. While at the Infrastructure and Compositions Layers, ubiquitous reuse of services is emphasised, components at the User Layer are not necessarily reusable by higher-level tools.

LifeWatch will provide a generic portal interface, which may be extended with domain- and application-specific portlets. However, it is quite likely that communities will develop user level tools appropriate to their particular circumstances. Such tools will provide restricted access to the LifeWatch capabilities but in doing so; the interface will be simplified and less susceptible to usage errors. These community tools may also need to be supported for specific client environments, such as embedded tools or PDAs. This approach lends itself to the addition of LifeWatch capabilities to existing tools with which a community's users may be more familiar.

4.3 Aspects of the LifeWatch architecture

4.3.1 The LifeWatch infrastructure as a Service Oriented Architecture

Figure 7 depicts an architectural vision of LifeWatch service components that is by no means exhaustive. These components have been related to the functional domain picture (Figure 6 above) by using the same colour encoding for the 4 layers: user, composition, infrastructure, and resource.

At the bottom of Figure 7, there are the resources, described above as resource layer. At the top of the figure are applications, which provide the interface of LifeWatch to the user. In the middle, there are several layers of service categories, which are the backbone for the LifeWatch infrastructure. The service bus represents the standardised mechanisms for connecting applications to the infrastructure as well as the services to each other.

The service layers are based on the ISO 19119 service classification, which is a key basis for grouping LifeWatch services. The boxes inside the layers represent a sub-categorisation of layers within the service taxonomy. Some of the boxes have been coloured in the same tone as the application and user layer (orange). This applies for thematic extension of service groups or services, to be used in a particular application domain. Generic service groups in Figure 7 are coloured white.

The service classification and the service categories are explained in more detail in section 6.5.

An additional service group, the source integration services, has been added to those defined by ISO 19119 due to the important role these play for the infrastructure. This group is settled directly over

the resources to indicate that resources have two interfaces: one conforming to the LifeWatch models to be used by other services within LifeWatch and one representing the proprietary characteristics of the resources outside the scope of LifeWatch (e.g. for use by foreign non-LifeWatch services). Source integration services provide an encapsulation of external resources to be used by the infrastructure.

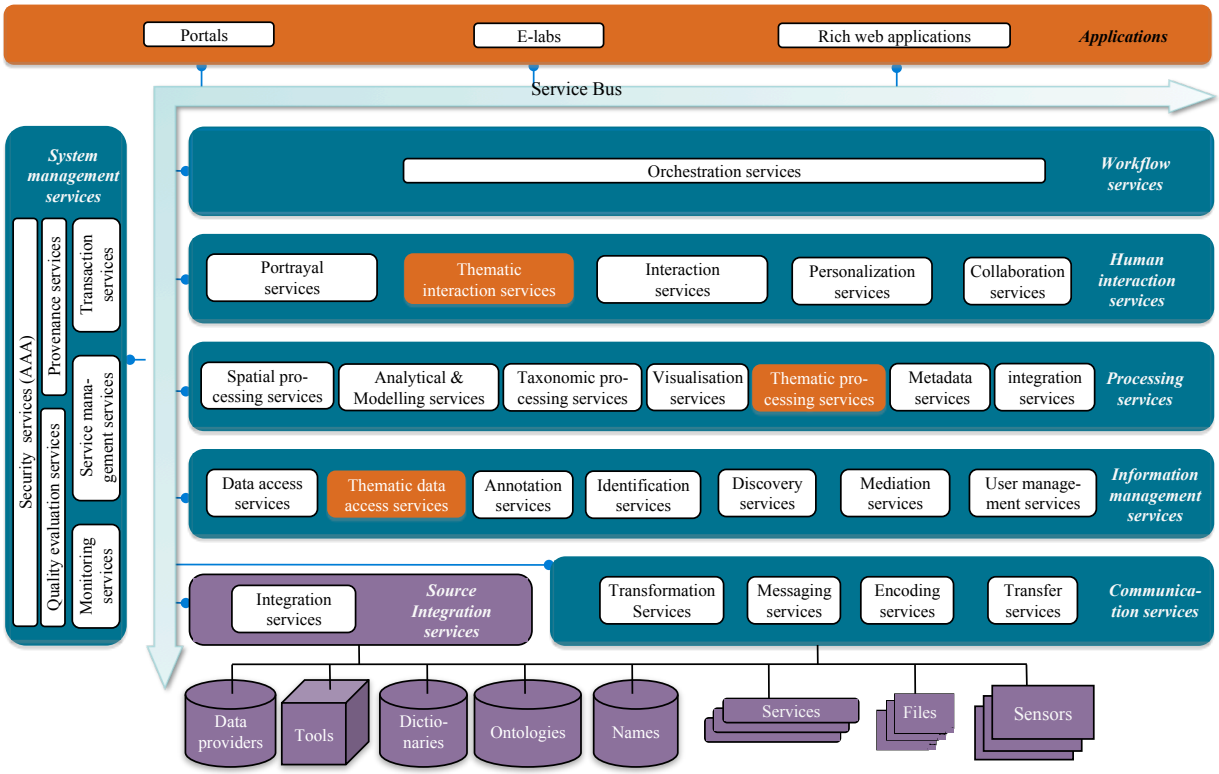


Figure 7: Service-oriented architecture

The layered service groupings indicate broad dependencies between services, with services from the higher layers using services from the same layer or from the layers beneath. For example, a processing service may call an access service to obtain input data, which may in turn call a transformation service to translate the data to the required format.

The System Management Services are orthogonal to the other service groups, as they are used at all horizontal layers and use or are used by the other services.

Services will be provided by many sources. Certain services such as the System Management Services are likely to be provided by an Infrastructure selected for LifeWatch while the stakeholders of LifeWatch will contribute others. Considering the heterogeneity of sources and activities, LifeWatch will provide a normative Reference Model to support achievement of interoperability between services.

4.3.2 The LifeWatch Reference Model

The LifeWatch Reference Model defined in the present document is a cornerstone for syntactic and semantic interoperability between all kinds of data and services.

The Reference Model lays down the rules for the specification of information models and services. Certain generic information models and meta-information models that are of general applicability for LifeWatch, and a number of core (basic) services that are generic for many application areas will be specified. Particular (meta-) information models and particular services may be added to cover specific "thematic" application areas within LifeWatch.

The choice and the definition of the Reference Model has been a key design decision in LifeWatch. The LifeWatch Reference model is based on the ORCHESTRA Reference Model that reflects many standards in distributed computing and geo-spatial information systems

The ORCHESTRA Reference Model defines two viewpoints, the Information and Services Viewpoints, as a platform-neutral specification of a distributed computing network. The Information Viewpoint specifies the information models to be used by the network. The Services Viewpoint specifies the functional operations occurring within the network in term of services.

Information models and service specifications come in two flavours - platform-neutral and platform-specific specifications. The platform-specific specifications (and of course implementations) must conform to the respective platform-neutral specification. This is a basic requirement for interoperability, at least for all information models and services that are of general interest for all of LifeWatch. LifeWatch will take advantage of the work done by ORCHESTRA in that the platform-neutral specification of many architectural services provided by ORCHESTRA will be (re-) used or exploited for modification, if necessary.

4.3.3 Semantic mediation for loosely coupled components

Semantic mediation will be another cornerstone for interoperability. It will be a differentiating feature of the LifeWatch approach. The need for semantic mediation arises in several areas:

- Data and Service Discovery: Discover data and services based on, for example, specific domain ontologies;
- Data Mediation: Processing data based on its semantics even if the data is provided by different data models;
- Data Fusion: Combining data from different sources;
- Data Interpretation: Multiple data models and heterogeneity at the data level itself, perhaps arising from differences of professional opinion, makes interpretation more challenging;
- Service Integration: Chaining of services often needs transformation of data when passing data from one service to another; and,
- Workflow identification: Discovery of workflows that, e.g., may help to solve a particular modelling problem.

Semantic mediation will be achieved by several means, presently under investigation. Mechanisms being considered include:

- Taxonomic checklists and associated tools for validating the integrity of checklists and cross-mapping between checklists32;
- Use of ontologies, including different ontology classes for different application domains (see 5.7 below);
- Use of semantic web innovations33; Use of rules-based reasoners. Once feature types (5.2 below) and ontologies are defined, it becomes possible to reason over the relations.

4.3.4 Service chaining as the basis for e-Science experiments

To the user, services provide "atomic" functionality. A biodiversity research task will usually involve a number of services that are executed sequentially, in parallel, or iteratively. LifeWatch will provide

---

[32] See, for example, the LITCHI tool: http://biodiversity.cs.cf.ac.uk/litchi/

[33] Such as the TUPELO semantic content management system: http://arxiv.org/pdf/0902.0744; http://dlt.ncsa.uiuc.edu/wiki/index.php/Overview

mechanisms for "service chaining" and speaks of a "workflow" when referring to a particular composition of service calls. Service chains can be programmed, defined in a script language or the user can compose them in a workflow editor. The latter are executed by a so called "workflow enactment engine". A portal may provide a user interface to a workflow so that the user interacts with a workflow by only pressing buttons and filling forms.

The most important aspect is that any such workflow is reproducible. In that sense, a workflow represents an experiment that is executed "in-silico". Any scientific result achieved through a workflow may come under scrutiny by other researchers or research groups.

LifeWatch will support the concept of workflows as "in-silico" experiments. Workflow editors and workflow enactment engines will be provided. Mechanisms for recording manually executed workflows will be considered as well.

The accentuation of workflows as a fundamental paradigm for biodiversity research conducted in the context of virtual e-Laboratories (i.e., a Virtual e-Research Environment (VRE) for biodiversity research) is another differentiating feature of the LifeWatch approach.

4.3.5 Unambiguous identification and descriptive mechanisms

The LifeWatch infrastructure must be able to identify each component and resource unequivocally. This is not only important for granting access to the correct resource after it has been discovered, but also for reproducing workflows and creating provenance information for each transaction in the life cycle of a resource in LifeWatch. LifeWatch will attach a Globally Unique Identifier (GUID) to each resource from the first moment a resource is used. There are currently several mechanisms for constructing and attaching a GUID and some sources already have such a property.

Life Science Identifiers (LSID) are one type of GUID that is presently proposed for uniquely identifying biodiversity resources. Such identifiers contain an authority identification field that uniquely identifies the source of the LSID. Often, this field is a domain name belonging to the organisation creating the LSID but there is provision for it to be allocated and maintained by TDWG should the need arise.

LifeWatch will investigate whether the LSID or another mechanism is most applicable for LifeWatch, taking into account the need to preserve identifiers that may have been assigned to objects prior to them “entering” the LifeWatch domain.

NOTE: Some users of LSIDs think that the concept of LSIDs is good but that the technical realization so far is not very easy to handle. The rigid rule for the need of versioning (for any change) is problematic. These aspects will be investigated further.

Source data can be changed due to improvements or intrinsic modifications. Consequently, workflows may produce new outputs and derived data may change. To cope with such data updates LifeWatch will use GUIDs with version numbers.

LifeWatch will need a registry (or catalogue) for all discoverable resources, attached to the GUID[34]. The catalogue will contain resources that are registered by a provider as well as resources that are automatically harvested. The ratio of holding catalogue information in LifeWatch and acquiring it on-demand is a trade-off between the flexibility and the performance of LifeWatch. This will be evaluated in the construction phase.

[34] One possibility is joint development with GBIF GBRDS (Global Biodiversity Resource Discovery System). This will be evaluated.

The catalogue will use a unified mechanism to obtain data (or meta-information), which will be searched by and provided to the user (or to a workflow service) during a discovery task. The unified mechanism requires that each LifeWatch resource is self-descriptive, at least through its access mechanism and/or semantic. LifeWatch will require that all services shall implement the Service Capabilities interface as a uniform access to the meta-information models describing the service and its resources.

Furthermore, LifeWatch will assist the discovery of resources by providing look-up mechanisms for vocabularies and code lists, using existing technologies, e.g. the generic thesaurus interface developed by the SYNTHESYS network. Users may want to select a service category, a species name, or a particular property type from existing lists. The lists may be specific for a thematic domain. LifeWatch will provide generic dictionaries as well as thematic ones, when appropriate, so that user can choose the proper dictionary.

4.3.6 Provenance

To make the sharing of data, methods and experiments in LifeWatch attractive, all data shall be traceable back to their origin and usage of source data shall be documented. The (re-) examination of scientific results demands an exact recording of workflows in terms of data and services. As noted above (3.6, composition requirements) the appropriate level of detail at which to record will need to be determined.

Change of source data conflicts with reproducibility. The versioning of resources reveals any changes. A key question for provenance is whether exact reproducibility must be guaranteed, and if so, who will hold all the older versions of the source data: the providers or the LifeWatch infrastructure. This issue will be assessed together with resource providers during the construction phase.

LifeWatch will provide automatic tracking as a provenance mechanism with the following capabilities:

- Estimation of data quality and reliability based on provenance information about the source data and transformations.
- Logging of Audit Trails: for instance, for determining resource usage and detecting errors in data generation.
- Replication: Data derivation may be reproduced and the currency of derived data may be maintained based on detailed provenance information.
- Attribution: Provenance can be used to establish the copyright and ownership of data, enable its citation, and determine liability in case of erroneous data.
- Informational: A generic use of lineage is to query based on lineage metadata for data discovery. It can also be browsed to provide a context to interpret data.

4.3.7 Platforms

With its platform-neutral specifications, the Reference Model is independent of a particular choice of platform and technology. LifeWatch will define platform-neutral interaction models as a reference for service interaction. This includes abstract models of how to organise service networks.

A platform in ORCHESTRA is comprised of: an interface language (such as WSDL); an execution context characterised by the transport protocol and the message format; by a security policy; and by machine readable service descriptions; further use of a schema language (such as ISO 19136::GML), a schema mapping (i.e. encoding rules for UML schemas), and constraints on the information models.

In the present era, services are often identified closely with Web Services but this may not always remain the case. Emerging paradigms such as cloud computing may be a future platform.

At present, LifeWatch will consider a combination of two platforms as the preferred technological basis: Web and Grid. For the Web, LifeWatch will take advantage of W3C and ORCHESTRA by re-using the latter's specifications of many services as Web Services. LifeWatch will take advantage of Open Grid Services Architecture (OGSA) by establishing itself on top of emerging European e-Infrastructures (e.g. EGI).

4.3.8 Data models

LifeWatch is about biodiversity and biodiversity related data. Since biodiversity includes four levels of organisation (genetic, species, ecosystems and landscape) and related data can range from meteorological parameters to measurements of human impact, no single data model is sufficient to organise this complex information. Several, possibly overlapping, models are needed to cover multiple application areas.

Nevertheless, the LifeWatch Reference Model promotes standardised data and attribute types in its models. The following data types can be regarded as generic since they are often used in the different biodiversity application models.

4.3.9 Geospatial data

Geospatial data contains information about the physical location of the data subject. It may be defined as a single point on the Earth's surface, or as an irregular region on the surface, e.g. the range of a species. Discovery of data with geospatial constraints can be computationally intensive, as the overlap of a search region with the regions from multiple data records must be computed. Complexity increases when the compared regions use different Coordinate Reference Systems. To simplify this matching, geospatial data standards commonly support the specification of bounding rectangles (known as bounding boxes) for collections of data. The matter becomes more complicated when geological locations and timescales have to be considered, as is the case in evolutionary and palaeontology-related climate change research. ABCD's Extension for Geosciences (ABCDEFG) provides some hints at the data domains to be included.

Geospatial data may have attributes, whose types are standardised by the ISO 19100 series: Spatial geometry and spatial topology describe the spatial position, extension and metric properties of spatial data, which makes the attributes comparable and usable by a great number of processing methods. Much biodiversity data does not yet have spatial attributes beyond the name of a spatial object (e.g. a place name or address). Through the use of geographic identifiers, such descriptions can be matched with spatial objects (e.g. via a gazetteer or geocoding service), which implies that a history of the naming and meaning of the name of special objects is to be kept (e.g. the name and meaning of the name of cities change over time).

Temporal attributes define a time stamp or a temporal range. Temporal ranges add an additional dimension to geospatially-referenced data, and increase the complexity of discovery. This is particularly true when geospatial values change over time.

4.3.10 Taxonomic data

When working with taxonomic data, it is important to know which model of taxonomy the data creator used. Taxonomic models have changed over time due to better understanding of the relationships between species. Multiple taxonomies can exist concurrently due to differing scientific opinions about species relationships. This means that the scientific name for a species can be ambiguous.

A Taxon Concept is a unique specification of a species, restricting the identity of a species to a single published definition. While this allows unambiguous identification of a specimen, it results in a complex set of overlapping relationships between the Taxon Concepts. As with the geospatial data, this makes discovery more computationally intensive.

4.3.11 Abiotic data

Much biodiversity data is used in conjunction with a wide range of abiotic data types describing the chemical and physical characteristics of the environment in which organisms are situated. Such data includes: topological data, climatologic data, soil chemistry data, hydrology data, temperature data (soil, air, sea), etc. to name but a few examples. Such data may occur in conjunction with relevant biotic data or may be generated and kept separately from it. Abiotic data usually has a geospatial component so discovery of relevant data poses similar problems to those described above.

4.3.12 Metadata

ISO 19115 and INSPIRE propose models for metadata that help to describe resources and provide quality information about those resources. ORCHESTRA does not specify meta-information attributes, but gives rules for the construction of meta-information models for specific purposes. Services should use these information models in their interfaces.

Metadata has a particular relevance for LifeWatch in order to support the description, discovery, and use of resources by LifeWatch users. However, the extent to which the abstract design of the infrastructure should only provide generic meta-information models versus comprehensive meta-models for biodiversity data is not yet fully clear. It requires further investigation.

4.3.13 Integration of Source Systems

There are data sources, systems and services which may already have existed for a long time and which pre-date the LifeWatch “era”. Often such systems are referred to as "legacy systems" but this is misleading insofar as the phrase “legacy systems” usually refers to systems that are no longer supported. This is not the case for these systems. Thus, we follow the terminology of ORCHESTRA and speak of source systems. A source system is a container of unstructured, semi-structured, or structured data and/or a provider of functions in terms of services. The source systems that LifeWatch will interact with are of a very heterogeneous nature and contain information of a variety of types and in a variety of formats.

A typical example of source systems for LifeWatch is biodiversity data collections. Data collections come in three variants at least:

- As existing data sources that are provided by dedicated data providers (e.g., natural history collections, monitoring organisations, etc.) or through data aggregators such as GBIF and OBIS;
- As new data, which may originate automatically from electronic sensor measurements or manually by recording from data collectors (e.g. human observers) through mobile devices or editing tools (captured data), and
- As data generated within LifeWatch, either through users who uploaded their data using the specified interfaces or data that was created by LifeWatch services, e.g. as results from integration, combination or transformation of data (LifeWatch data).

Systems constructed solely in pursuit of LifeWatch objectives and complying fully with the architectural constructs of LifeWatch can be referred to as LifeWatch Source Systems. However, many pre-existing systems, despite having significant relevance for LifeWatch, have been constructed primarily for other purposes and using architectural concepts outside of LifeWatch. These are referred to as External Source Systems. Such systems serve multiple domains and often play an important role in the community as a whole (e.g., GBIF systems)

LifeWatch faces a significant challenge to accommodate such External Source Systems and to make their content accessible from the LifeWatch infrastructure. The general idea is to provide External Source Systems with interfaces that conform to LifeWatch principles. With these interfaces provided an External

Source System is transformed to become a LifeWatch Source System. For the designer of a LifeWatch platform, transforming an External Source System into a LifeWatch Source System is a major task, which is called source system integration.

There are several strategies LifeWatch shall pursue:

- Conversion services perform the transformation from one format to another known by LifeWatch
- Use of standard protocols for the transfer of data, which are independent of the requested data models (e.g. TAPIR protocol),
- Conversion of models to supported schemas (e.g. transforming ABCD or Darwin Core to GML).
- Semantic mediation between different formats, or/and semantic systems (taxonomies, name-spaces)
- Extraction of all relevant meta-information for using the data with LifeWatch services.

For capture data, one would expect that standardised services and protocols (e.g. the OGC Sensor Web Enablement suite of standards, GEOSS and GMES services) would be a part of the architectural services provided by LifeWatch.

Source system integration consists of two aspects: Firstly, the practical needs of and constraints on Data/Service Providers must be well understood by LifeWatch. Secondly, taking these needs and constraints into account, LifeWatch must be able to provide a clear and unambiguous statement of the requirements that a Data/Service Provider must meet in order to be "admitted" to the LifeWatch infrastructure.

To become associated with LifeWatch, Data/Service Providers will need to commit to a "Service Level Description" (SLD), detailing their provision of data resources or services and any constraints on the use of those resources. While it is important not to discourage eager participants from joining, LifeWatch must ensure that participants can contribute meaningfully to an integrated system. Hence, the SLD template must be both rigorous and flexible in order to handle the Data/Service Providers' differing expectations and capabilities successfully.

A series of procedures for "admitting" (i.e., signing up, integrating and testing) a Data/Service Provider to the LifeWatch infrastructure will also be developed. The SLD template can be tailored to the particular requirements of an individual Provider as part of the admission procedures. The template will provide potential Providers with a detailed description of the benefits and commitments required for them to participate in the LifeWatch infrastructure. A Provider can be subsequently monitored against the requirements of the agreed SLD to ensure that the high standards for LifeWatch data and service quality are maintained throughout the operational phase.

5 Information Viewpoint

5.1 Overview

The information viewpoint provides the specification framework for all categories of information handled by the LifeWatch Infrastructure. This framework does not specify a specific information model, but will be the basis for constructing the LifeWatch Conceptual Models based on object-oriented paradigms.

The LifeWatch Information Framework distinguishes the following levels, which will be discussed in detail below:

- Source system level
- Feature level
- Application Schema level
- Semantic level.
- Meta-model level

The source system level comprises all resources containing relevant data or providing functional services. This level is equivalent to the resource layer at the LifeWatch architecture (Figure 7). Examples are service providers, catalogues, and databases.

The feature level abstracts the physical data and services provided by the resource level by representing their contents in terms of features. Features (ISO 19109) are the fundamental units of information handled by the LifeWatch infrastructure.

On application schema level, the logical structure of the presentation of data and services is defined. An application schema addresses the logical organisation, rather than the physical (ISO 19109). It provides a model of the universe of discourse, i.e. all the entities of the real world with their attributes and the association between them, which are needed to fulfil all the capabilities required in LifeWatch.

The meta-model level provides the rules to define the application schemas.

The semantic level covers the semantics of the information specified on the other levels in terms of ontologies or similar to be exploited and the purposes for which such ontologies are to be applied to achieve semantic interoperability.

5.2 On information models

Information models are specific for an application (area), allowing the integration of data and services of existing systems and the usage of ontologies. The ORCHESTRA modelling approach for content information distinguishes between:

- General rules to be applied to application schemas, defined by a meta-model,
- Application schemas based on a meta-model, which define the application structure and data types (usually referred to as the application model), and
- The content information or data, which are accessed through services according to the application schema (usually referred to as the data model).

The ISO 191xx series establishes a structured set of standards for information concerning objects or phenomena that are directly or indirectly associated with a location relative to the Earth (ISO 19101). This standard specifies methods, services, and tools for management of geographic information, including the definition, acquisition, analysis, access, presentation, and transfer of such data between different users,

systems, and locations. The ISO geographic information series of standards is flexible allowing a large number of options that may be tailored to suit any application.

LifeWatch application and data models rely on a fundamental concept of the ISO 191xx series: the feature. A feature is defined as an abstraction of a real world phenomenon. A feature type35 is used to group features having similar characteristics.

Example: "Tower Bridge" is a feature, the abstraction of a particular real world bridge. The term "bridge" is the abstraction of the collection of all real world entities that are classified into the concept behind the term "bridge". This is the feature type. Given an application domain that deals with "bridges" and "roads", a feature type "Bridge" or "Road" specifies the abstract properties of the respective collections whilst a named feature instance (Tower Bridge) is an element of this collection. The set of all features (all bridges and roads in this example) should be thought of as a representation of the real world. ❑

In the world of LifeWatch, features represent the biodiversity content information36. Features are the data objects of LifeWatch. In LifeWatch, feature types such as "OccurrenceRecordType" or "TaxonType" are abstractions of real world phenomena from the domain of biodiversity research.

Figure 8 illustrates the ideas expressed above.

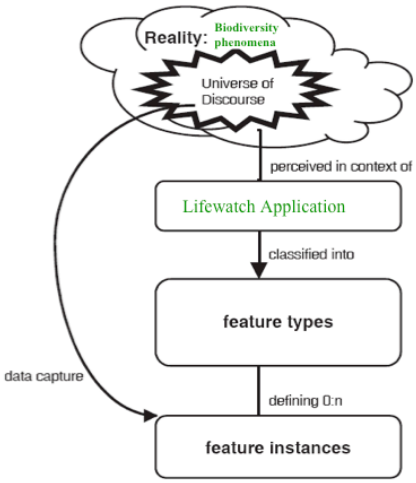


Figure 8: From reality phenomena to feature instances (Source: ISO 19109)

The universe of discourse is the view of the real world and everything in it of interest from the point of view of the LifeWatch stakeholders i.e., the biodiversity research domain.

Rules of the meta-model are used to define an application schema based on the defined feature types. According to ISO 19109, "an application schema defines the logical structure of data and may define operations that can be performed on or with data. An application schema addresses the logical organisation, rather than the physical." Similar to INSPIRE [INSPIRE, p. 39], LifeWatch defines an application schema for the generic LifeWatch feature types and their properties. We refer to this as the Base Application Schema. The Base Application Schema is meant to subsume all the feature types and their relations independently of a particular application. Relations between feature types are defined in terms feature association types and inheritance. Properties of feature types are feature attributes, feature operations, and feature association roles. In LifeWatch, the Base application Schema constitutes a "back plane" for all

35 In an object-oriented programming paradigm a class is equivalent to the idea of a feature type.

36 Existing biodiversity models often use terms like “concept”, “unit”, “taxon” and “parameter” to name content.

LifeWatch applications. Other application schemas may extend or restrict the Base Application Schema. They are referred to as thematic application schemas.

The conceptual schema language for the Base Application Schema and other platform-neutral schemas is UML. Otherwise, a platform-specific language (e.g., XML, RDF, OWL) is used.

NOTE: The distinction between features and attributes depends on semantics. As a general rule, a particular piece of information will be considered as a feature (type) if it represents a concept of particular importance with an identity of its own, such as an abstraction of a "real world" entity.

In contrast, a concept will be modelled as an attribute if it is an auxiliary concept with no identity of its own, which is used only in the context of a feature such as, for instance, temporal or spatial information associated with a feature.

5.3 The LifeWatch information framework

The LifeWatch Information Framework distinguishes between data and information (following the Federal Standard 1037C37):

- Data: Representation of measurements, facts, concepts, or instructions in a formalised manner suitable for communication, interpretation, or processing by humans or by automatic means.
- Information: Meaning assigned to data by means of the known conventions used in their representation. The meaning that a human assigns to data by means of the known conventions used in their representation.

The "known conventions" are those specified by LifeWatch. The distinction between data and information is important in that it distinguishes between data being entities provided and specified outside of the scope of LifeWatch while information is the interpretation and representation of these data within the scope of LifeWatch. To give an example: an occurrence record for a species "Felis silvestris" is "data ". The representation of the occurrence record as a feature with a specific feature type specified within LifeWatch (i.e. "occurrence_record") with its known attributes and relationships, defined by a model, is "information".

The LifeWatch Information Framework further distinguishes between two aspects of information:

- Primary and derived information (including metadata) related to biodiversity research
- Meta-information, that is: descriptive information about available information and resources with regard to a particular purpose (i.e. a particular mode of usage)

The definition of meta-information stresses that the need for meta-information arises from particular tasks or purposes (like catalogue organisation or data storage), where many different services and data objects must be handled by common methods and therefore should have common attributes and descriptions. For example, for the purpose "Discovery", related to catalogue organisation, attributes such as: "taxonomic_species", "spatial_location", and "time_frame" are the only attributes used, while other properties of a feature of type "occurrence_record" like "date_of_creation" may be irrelevant for that purpose. In that meta-information reflects only a part of or may be extracted from the information of a data object. Meta-information is purpose or context dependent in that what is meta-information in one context may be information in another context and vice versa.

---

[37] Federal Standard Telecommunications: Terms and definitions extracted from Joint Pub 1-02 (DOD Joint Staff Publication No. 1-02), 1994, *Department of Defense Dictionary of Military and Associated Terms*, http://www.its.bldrdoc.gov/fs-1037

Meta-information should not be confused with the more common “meta-data” which is "data about other data“. For instance, metadata regarding a book would be the title, author, and date of publication, subject, a unique identifier, its dimensions, number of pages, and the language of the text with the actual text being the “data“. These “meta-data“ will be considered as “meta-information“ for discovery of the book in a library, namely for detecting the library specific index number guiding to the actual position on the library shelves. Given the index number these “meta-data“ become irrelevant for the purpose “Access“, hence is “information“ in this context.

Note that different meta-information models are used - one for each particular purpose. Of course, one might take the sum of all the meta-information to provide one meta-information model but that would unduly restrict flexibility in that any piece of information has to be complemented by an unnecessarily great number of meta-information details that are useless for the purposes or tasks that apply to the particular piece of information.

At present, the following purposes have been identified and will be supported by meta-information models in LifeWatch:

- Discovery
- Identification
- Access, Storage and Invocation
- Integration
- Orchestration
- Personalisation
- Quality evaluation
- Provenance
- Authentication, Authorisation and Accounting (AAA)
- Data capture (sensors and mobile devices)
- Collaboration
- Human Interaction

A description of these LifeWatch purposes is given in B.5.

The components of the LifeWatch Information Framework and their reference to the different levels and aspects are captured in Figure 9 that is based on the ORCHESTRA Information Model Framework (RM-OA V2, Section 8.6.2).

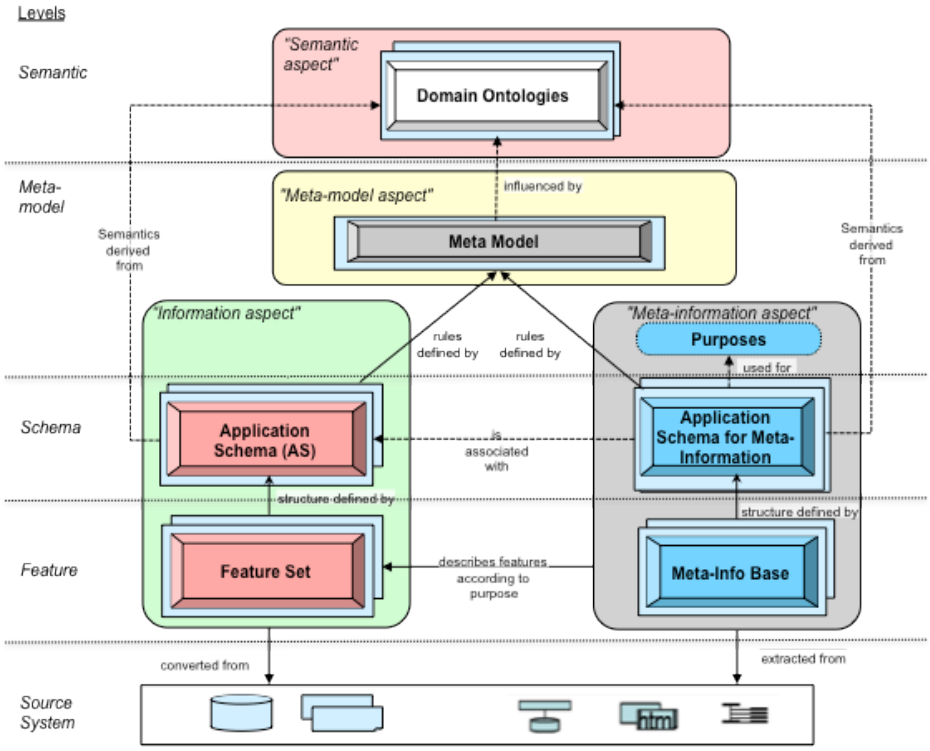


Figure 9: The LifeWatch Information Framework

The picture is divided into several parts. The six informational levels mentioned before are on the very left.

The "information aspect" is concerned with the information related to the biodiversity domain. The domain ontologies provide semantics to the information specified on the other levels. Several ontologies may coexist. They are defined and shared by the user communities. The Biodiversity feature set subsumes all the biodiversity information represented as features according to the application schemas.

The "meta-information aspect" is concerned with the meta-information models. The ORCHESTRA Architecture does not define "the" single meta-information model valid for any purpose, but allows several meta-information models, each of which reflects a particular purpose. The RM_OA specifies a set of rules to be followed for particular purposes such as "Discovery", "Access, Storage, and Service Invocation". The structure of meta-information is defined by Application Schemas for Meta-Information Models (LifeWatch AS-MI). The Meta-Info Base is the store for meta-information. It contains information that describes features in the form of the Biodiversity Feature Set according to a well-defined purpose.

It is worth noting that purposes, hence meta-information models, reflect more or less an ICT perspective on the LifeWatch Architecture while the "information aspect" reflects the biodiversity domain, hence the user perspective.

The "meta-model aspect" captures all the normative rules to be applied when constructing the entities on the other levels.

5.4 Meta-model level

The LifeWatch Meta-Model constitutes the common framework for all feature-based application schemas defined within LifeWatch. It defines all the rules for the specification of a LifeWatch application schema. The LifeWatch Meta-Model builds on the ORCHESTRA Meta-Model (OMM). The OMM is an evolu-

tion of the General Feature Model (GFM) of ISO 19109 in that it defines a subgroup of GFM classes and properties relevant to ORCHESTRA and adds additional meta-classes for meta-feature and attribute types. The meta-feature types added are a Document Descriptor Type and a Coverage Type that are eminent generic information types. LifeWatch may add other generic meta-feature types of particular relevance for the biodiversity domain (see Figure 10). However, the addition of predefined feature types should be handled carefully so as not to migrate too much domain specific information to the meta-level which might unduly constrain the definition of application schemas.

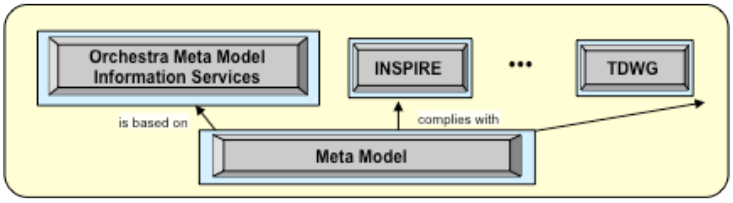


Figure 10: Compliance of the LifeWatch Meta-Model to Standards

Data types are comprised of Basic Data Types which have a standardised definition by a standardisation body (e.g. ISO 191xx series), ORCHESTRA predefined (meta) types, and LifeWatch predefined (meta) types, and user-defined types. All data types used and defined in LifeWatch, including the predefined types, shall be built directly or indirectly (using predefined types) from the Basic Data Types. All the definition and rules of the ORCHESTRA Reference Model apply.

The components of the LifeWatch Meta-Model are outlined in Appendix B.

5.5 Application schema level

The application rules define the steps to follow and the requirements to satisfy when defining an application scheme. The general scheme is to develop a conceptual model based on the requirements from the intended field of application for identification of feature types, a description of the feature types in UML (for the abstract level) and/or in platform-specific language using and complying with standardised schemas such as spatial or quality schemas. The respective rules will be specified in Appendix E.

Application schemas are to be defined for the particular application areas in LifeWatch (e.g. collections, niche modelling) according to the rules of the LifeWatch Meta-Model. An application schema should have normative character for the particular application area, implying that all involved groups agree on the application scheme.

There will be one distinguished application scheme, the LifeWatch Base Application Schema that is meant to provide a uniform view of those entities that are relevant for all of LifeWatch. Such entities may be taxonomies, taxon occurrence records, sensor data, etc. Identification of these entities will take place during the preparation phase. Additionally, for each of the identified entities, a proposal for an application scheme will be made in co-operation with the stakeholders. Definition of other application schemas will take place during the construction phase.

Note: The Reference Model can only outline strategies of how to approach the definition of the Base Application Schema trying to cover the design space and the design options. This will be done in the construction phase. Further stipulations need discussion with experts of the biodiversity informatics field. Here taxon occurrences are used as running example.

Example: Container Model for Taxon Occurrences

There are two extremes to accommodate taxon occurrences as a Base Application Scheme:

- To define one common model.
- To allow for usage of all models, like ABCD or DarwinCore.

Considering the efforts related to standardising formats in the area, agreement on a common model is unlikely for the near future; while the obvious benefit is that all services related to taxon occurrence records can rely on the format and content of such records.

The alternative is to have a container type that allows interpretation of a feature according to a context. The container may hold data of type "LW_TaxonOccurrence" that is the aggregation of all data models used for taxon occurrences (Figure 11). The context then indicates which data model is actually used.

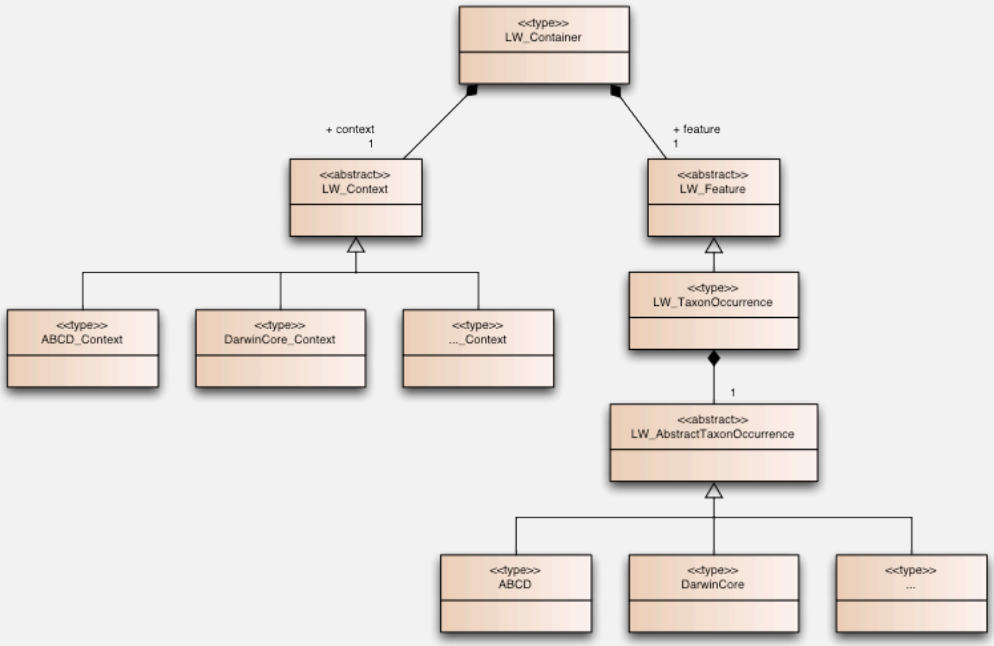


Figure 11: Container model for taxon occurrences

The idea is that the context provides sufficient information for accessing and updating the taxon occurrence data contained. For instance, if one wants to access spatial information (where did the taxon occur), the context indicates which data model is used, hence which kind of spatial information is available, and further which services may be used to extract the spatial information. A problem is that the different data models may use different notions of spatial information. Agreeing on a particular data model for spatial information may seem to be as difficult as to agree on a data model for taxon occurrences, but fortunately, there are standards (e.g. ISO19107::GM_Object). Thus, extraction of spatial information for different data models for taxon occurrence can result in data of the same attribute type.

Now one could require that, whenever a data model has a component modelling spatial information, this component must be conformant with standards, here "ISO19107::GM_Object". This strategy appears as too restrictive since it might exclude data models presently used in biodiversity research.

At present, the best overall strategy appears to be to accommodate both:

1. To embed existing data models and services by a strategy using containers (or similar), and
2. To require of LifeWatch Application Schemas that information models use standard attribute or feature types (or subtypes of these) whenever feasible.

5.6 Source system level

LifeWatch aims to integrate or link information from different data and service providers. Those providers may already exist (e.g. GBIF, EurOBIS, CoL) or may be implemented as LifeWatch Nodes using service implementations based on the LifeWatch specifications. The information framework is a mean of achieving syntactic and semantic interoperability between the resources underlying the infrastructure network. All those providers, as well as the different storage system for specific LifeWatch information (support information) like catalogue registries, temporal data, etc. constitutes the LifeWatch repository and represents the source perspective of the infrastructure.

The process of converting information from the repository to be used at the data level will be carried out by services (e.g. feature access services). LifeWatch will access the information through well-known interfaces (protocols). The specific format used by the protocol can anytime be extended or wrapped to conform a required model. The process of extracting meta-data from the repository will be differentiated from the one of extracting the data and will be specified on the context of the definition of meta-information services to be applied for a specific purpose.

Example: Catalogue Service

This service can use different application schemas for the refinement of meta-information according to the service type or the thematic context. Figure 12below illustrates an example of ad-hoc binding for the purposes "discovery" and "access".

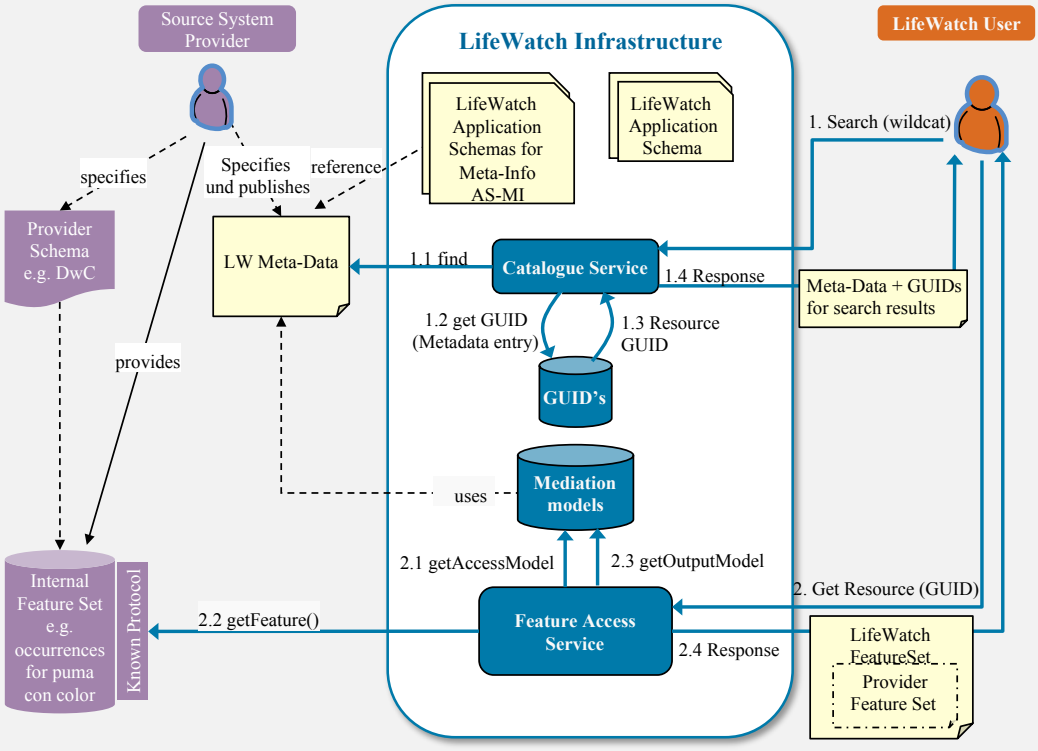


Figure 12: Discovery and access of ad-hoc published features

Source system provider for occurrences data, holding and providing the data in a particular schema, its data can become visible for LifeWatch by complying or specifying metadata according to one or more LifeWatch Application Schemas for Meta-Information (LW AS-MI). This data can be stored in own catalogues or be published to a LifeWatch catalogue repository. If a user searches the LifeWatch catalogue for a "taxon occurrence record" of, e.g., "Felis silvestris", the LifeWatch catalogue can find the resource

meta-data in catalogue-resources, which may be either integrated in the infrastructure or provided by a remote network. The resource must have a unique identification, which will be given or accepted by the LifeWatch Infrastructure the first time the resource (or its metadata) is used by a LifeWatch service. The user gets the LifeWatch meta-data for "Felis silvestris" and can request the data via an access service using the identifier. The access service uses the meta-data to locate the service, extracts the information from the source provider using a specified protocol, and applies mediation methods to transform the data into the LifeWatch Feature Set. Alternatively, it can provide the data using elements of the LifeWatch Application Schema as a wrapper, encapsulating the provider schema.

5.7 Semantic level

The semantic level allows LifeWatch to apply mechanisms for providing structures to manage the meaning of information. An example of such a mechanism is the use of ontologies, defined and used by the different communities related to biodiversity research. Ontologies provide a shared vocabulary, which can be used to model a domain. That is, the type of objects and/or concepts that exist, and their properties and relations. It can be used to reason, in other words: to be processible by machines, which can infer from the rules, allowing as far as possible, an automatically supported data and services interoperability.

ORCHESTRA defines different ontology classes for different application domains. Those are: (See RM-OA V.2, Chapter 8.6)

1. Domain Ontology: Provides a source of pre-defined concepts for a subject area (domain) such as ecology, biology, or genetics. Domain ontologies should provide a semantic reference for (meta-9 information models.
   Domain ontologies should coordinate with task ontologies. Both, the domain and task ontologies are decoupled from implementation concerns and provide a semantic reference to the information models and meta-information models.
2. Task Ontology: Formalises the knowledge related a specific problem or task, abstracting the level of a specific situation or organisational context. A task can be the monitoring of biodiversity changes, which can be applied in the domain or ecological or biological domain. Task ontologies can be used for workflow modelling.
3. Application Ontology: Codifies knowledge related to complete a task in a specific situation and organisational setting (e.g. the monitoring of biodiversity changes performed by the World Conservation Monitoring Centre at the United Nations Environment Programme UNEP). Application ontologies contain little reusable knowledge by other organisations and serve to "provide a semantic interface between the domain and task ontologies in the application" in case that a domain or task ontology cannot be mapped directly to the application context. Application and data ontologies can be used to support the integration of source systems, e.g. for schema mapping, for extracting meta-information and information and for the translation between different exchange formats.
4. Data or Service Ontology: "Describe a service or data source and may be seen as a special type of an application ontology". Service ontologies may also be used to support the integration of service providers with a focus on the discovery and mediated access to the services.

The ontologies appear on two levels depending on the development stage:

- *Conceptual ontologies* built by domain experts, based on their understanding of the domain
- *Logical ontologies* created through the transformation of conceptual ontologies into machine-readable notation.

Ontologies will be used for semantic annotation of all entities within the LifeWatch framework: (meta) information models, data sets, services, workflows, etc.

Many source providers and user communities, for which LifeWatch will be a research platform have developed or are working on the development of ontologies. Generic concepts that are relevant across the domains and which are canonised by a "high-level ontology" may generate possible candidates for additional meta-classes in the LifeWatch Reference Model. Similarly, if ontologies have a sufficient level of detail, they may induce the definition of an application scheme.

It is also possible that several ontologies may coexist; even within a thematic context. Users (or an administrator) may select which ontology should be used for mediation within a defined application or task.

LifeWatch promotes a modular hierarchical approach to the specification of ontologies implying that more complex ontologies will be composed from simpler established ontologies and that ontologies come at different levels of granularity relating concepts of more abstract or coarser level to domain ontologies which capture more specific knowledge at a finer level of granularity.

Besides ontologies, other mediation mechanisms can be used. LifeWatch should at least provide mediation mechanisms for taxonomies, named places and their conversion to geographic coordinates, for data and protocols interoperability, and for the classification of processing services in order to facilitate service chaining through workflows. Examples are synonym lists, gazetteers, and taxonomic checklists.

5.8 An application schema example

Figure 13 shows a simplified example of an UML model for a "SpecimenOrObservationBase" feature type derived from the SpecimentOrObservationBase class of the Common Data Model (CDM) of the EDIT Platform for Cybertaxonomy38. The schema modifies the CDM definition by omitting class-relationships (which are written as data types besides the attributes (e.g. Stage)), omitting specialization classes from SpecimentOrObservationBase, and modifying temporal and spatial attributes according to the Meta-Model rules (e.g. introducing OMM_TemporalAttributeType for the EventBase type).

[38] See EDIT: European Distributed Institute of Taxonomy (http://www.e-taxonomy.eu/) and the CommonDataModel (CDM) v1.4 http://dev.e-taxonomy.eu/trac/wiki/CommonDataModel

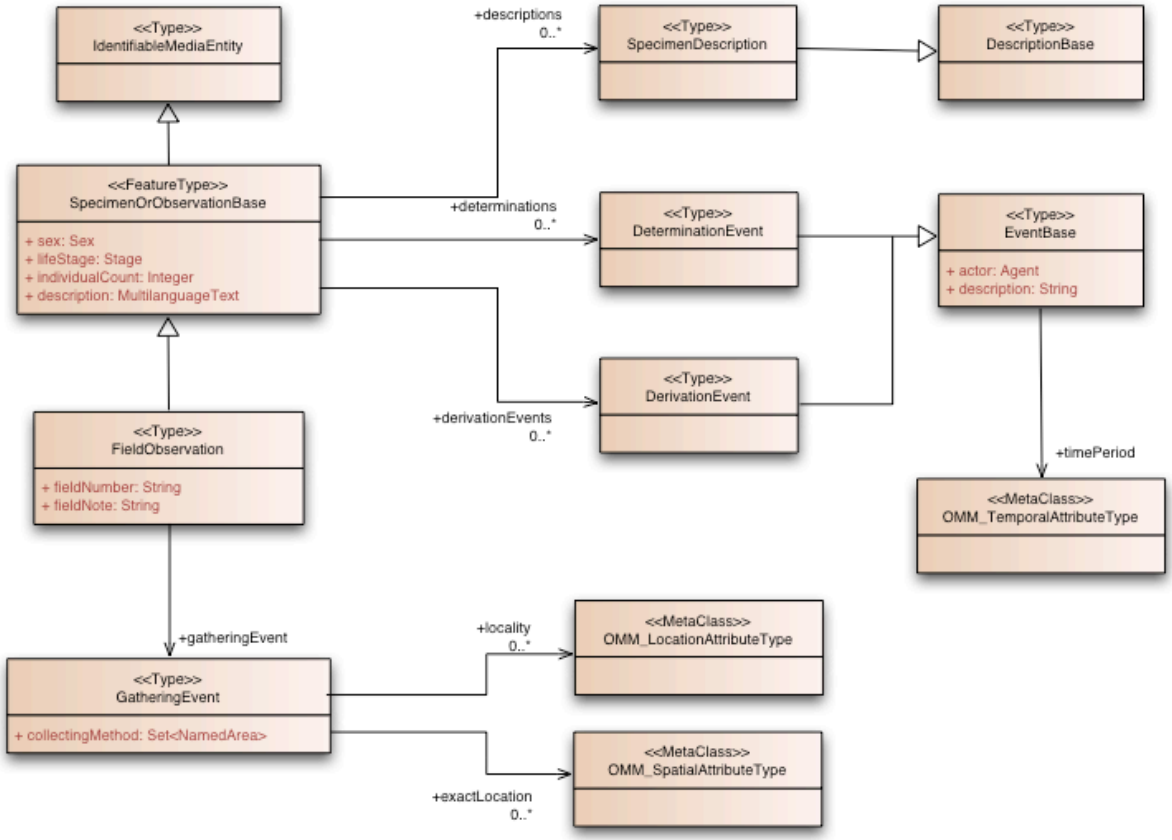


Figure 13: Example schema of a feature type definition for SpecimenOrObservationBase

Blank page

6 Service Viewpoint

6.1 Introduction

Following the ODP Reference Model, the computational viewpoint describes the functional decomposition of the infrastructure into distributed objects that interact at interfaces in an abstract, implementation, and platform neutral manner. The LifeWatch Infrastructure, as a distributed environment, refers to services as its basic objects. Therefore, this section is named Service Viewpoint.

The Services Viewpoint provides a framework for the development of specifications of LifeWatch services. It abstracts from implementation specific aspects like different architectural styles such Message Oriented, an Activity Oriented, or a Resource Oriented style (see the Web Services section of the Status Report on Infrastructures for Biodiversity Research[39]) but provides common semantics and basic principles of service design so that can be used across and between different architectural styles and implementations.

The Services Viewpoint relates the ORCHESTRA Service Model to the OASIS (draft) Reference Model for Service Oriented Architectures40OASIS-SOA-RA. The first section recapitulates basic concepts of service-oriented architectures as defined by the OASIS Reference Model for Service Oriented Architecture OASIS-SOA-RM as a guideline. On this basis the LifeWatch Service Model is introduced in Section 6.3. To facilitate the description and discovery of services, ISO 19119:2005 recommends the definition of "well-known service types" that are published for potential consumers through abstract or platform-specific service specifications. Thus, service types, as generic names, are part of service identification and may be classified according to different aspects for providing the taxonomy of services offered by the infrastructure, as described in section 6.5. Chaining of services provides added value to service oriented designs. The LifeWatch Reference Model uses the term workflow as a generic concept for service chaining. The core aspects of service chaining are described on section 6.6.

6.2 Basic concepts of service-oriented architectures

OASIS-SOA-RM defines Service Oriented Architecture (SOA) as a "paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains". The person or organization offering a capability is called a Service Provider and the entity having a need to be solved by one or more capabilities is called a Service Consumer.

The key concepts attached to a Service are:

Visibility	**Exposure of capabilities to be found by an entity with needs, typically done through Service Descriptions**
Interaction	The activity of using a capability, typically through the exchange of Messages between consumer and provider in a particular Execution Context
(Real World) Effect	Purpose of using a capability, result of an interaction.

[39] LifeWatch Working Document "Infrastructures for Biodiversity Research – Status Report, May 2009

[40] OASIS-SOA-RM: OASIS (2008): Reference Architecture for Service Oriented Architecture Version 1.0, 23 April 208, OASIS authoritative document. Available under http://docs.oasis-open.org/soa-rm/soa-ra/v1.0/soa-ra.pdf (August 2009)

To achieve visibility between service consumers and a service three preconditions must be fulfilled:

- Potential service consumer must know the existence of the service. This is known as Awareness. Service Discovery mechanisms are intended to provide service awareness, allowing the access to service descriptions and policies.
- The service consumer and service provider must engage to interact with each other, according to the service policies. This is only possible if the service provider is willing and does not deny the service to the consumer.
- There must be a communication path for the exchange of messages between service consumer and provider, which means that the service must be reachable for the consumer. A service is reachable through the exposed and available service endpoint, specified in the service descriptions.

Service Descriptions contains the necessary information to facilitate visibility and interaction with a service in terms of inputs, outputs, associated semantics, as well as conditions for using the service. They allow potential service consumers to decide if the service is suitable for their needs and if their context satisfies the requirements of the service provider. How a service must be described depends on the usage purpose (context and the particular needs of service consumers). A service must describe the service functionalities, the information models and the (purpose-oriented) meta-information models involved in service interactions, the Policies related to the services (constraints or conditions on the use, deployment, or description as defined either by service providers or by a service consumer), and the Contracts (agreement between two or more parties). The service functionalities (operations) and the related (meta-) information models are specified through the service interfaces.

Service Interaction occurs in general by sending and receiving messages between service consumer and the service. The OASIS-SOA-RM uses the terms Information model and Behaviour model, which form part of the service description, as the concepts on which service interactions are based on. The information model describes the structure (format, relationships, and the vocabulary) of the information that may be exchanged with the service. The behaviour model provides the actions, which may be invoked against a service and implied effect of the actions, as well as the process or temporal dependencies related to the interaction. The actions may be unrelated to the service functionalities, for example to present credentials for authentication, or may define action sequences (conversations), e.g. to perform a transaction operation. The temporal relationships of actions and events associated to a service interaction refer to properties of the service or actions, including if they are idempotent, long-running, transaction etc. They also allow the definition of orchestrations and choreographies involving the service.

The Execution Context of a service interaction is the set of infrastructure elements (components, policies assertions and agreements) identified as part of an instantiated service interaction, forming a communication channel between service consumers and provides. The execution context can be created dynamically, e.g. when intermediary elements must be included for encryption, translation, or transformation of formats and technologies or static like the so-called Service Bus, which provides means for interoperability between services based on predefined protocols and communication mechanisms.

The effect of a service interaction is often the change of a so-called shared-state, which refers to the explicit visible state for consumer and provider. The elements of the shared state should be inferable from the a priori interaction.

6.3 The LifeWatch service model

The LifeWatch Service Model provides the elements and rules for the specification of LifeWatch services to achieve syntactic and semantic interoperability between services, source systems, and applications. The service model conforms to the basic SOA concepts as defined above and it relates the OASIS Reference Model and the ORCHESTRA Meta-Model for Services.

LifeWatch Service is the metaphor for the functional entities acting in the LifeWatch infrastructure and has two different appearances, namely as

- Service components - the software components that implement the interfaces specified by the service types, and
- Service instances - running instances of a service component in a service network.

LifeWatch services are instances of

- Service types – as service schemas that are specified in terms of the externally visible behaviour.

The Service Framework in Figure 14adapts the ORCHESTRA Framework for Services [RM-OA] and distinguishes levels and components similar to the Information Model in Figure 9:

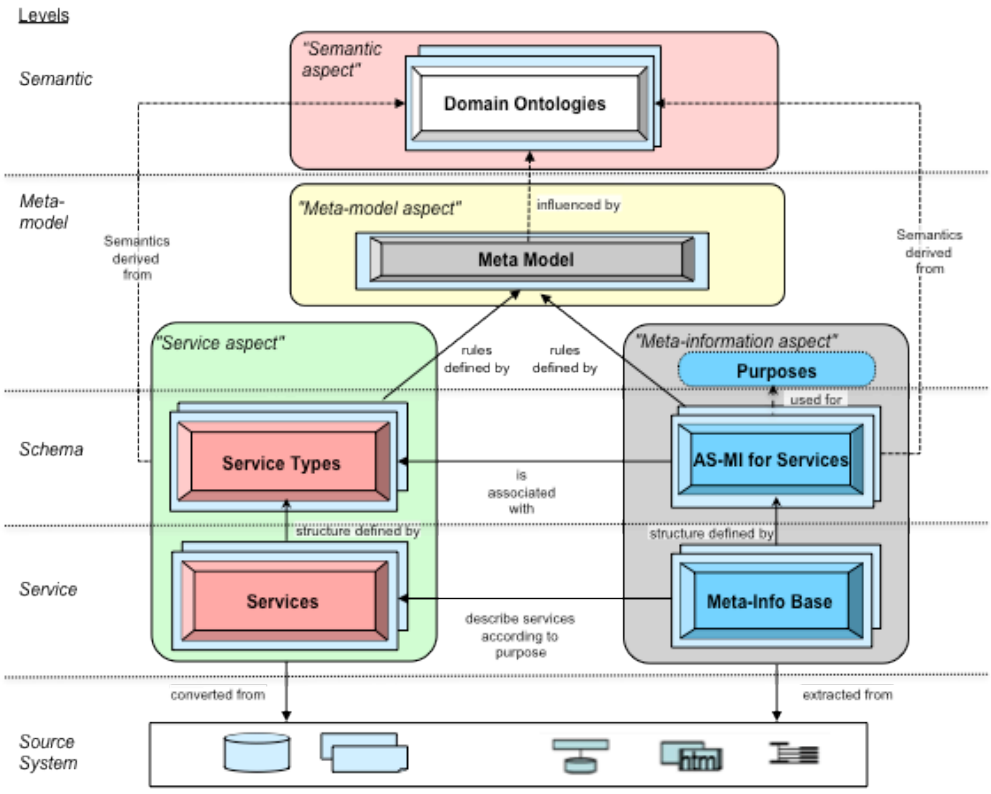


Figure 14: Framework for LifeWatch Services

The meta-model level provides the basic rules for construction of service specifications in terms of a Service Meta-Model. The LifeWatch Service Meta-Model extends the ORCHESTRA Service Meta-Model by defining LifeWatch specific rules and concepts, to be used by the LifeWatch service types and meta-information models (see Appendix C).

At the schema level, service types and associated meta-information schemes are defined. Service types are specified in terms of interfaces and of meta-information, both of which reflect their external visible behaviour. The Application Schemes of Meta-Information for Services (AS-MI) describe the structure of the meta-information associated to service types with regard to particular purposes.

The service level consists of all the LifeWatch Services and a Meta-Info Base that holds all the meta-information about services. Meta-information again relates to purposes. For instance, for “Discovery” of services the meta-information may specify which interfaces are offered. For a semantic search, it may specify which functionalities certain operations have. For a purpose “Orchestration”, the input/output

types of operations may be of interest or it may be of interest whether the service is a translation service (a so-called “shim” service).

The source system level represents the access to services outside of the LifeWatch framework. There is no specific service type for source systems, but source system (service) components should reuse the interface types supplied by the LifeWatch Reference Model.

The semantic level refers to the mechanism to achieve semantic interoperability, expressing the domain knowledge in form of descriptions of concept and their relationship to be used with the description of services and meta-models. They can be expressed in ontologies languages, so they can be provided to the service consumers.

The Service Framework is refined in Figure 15.

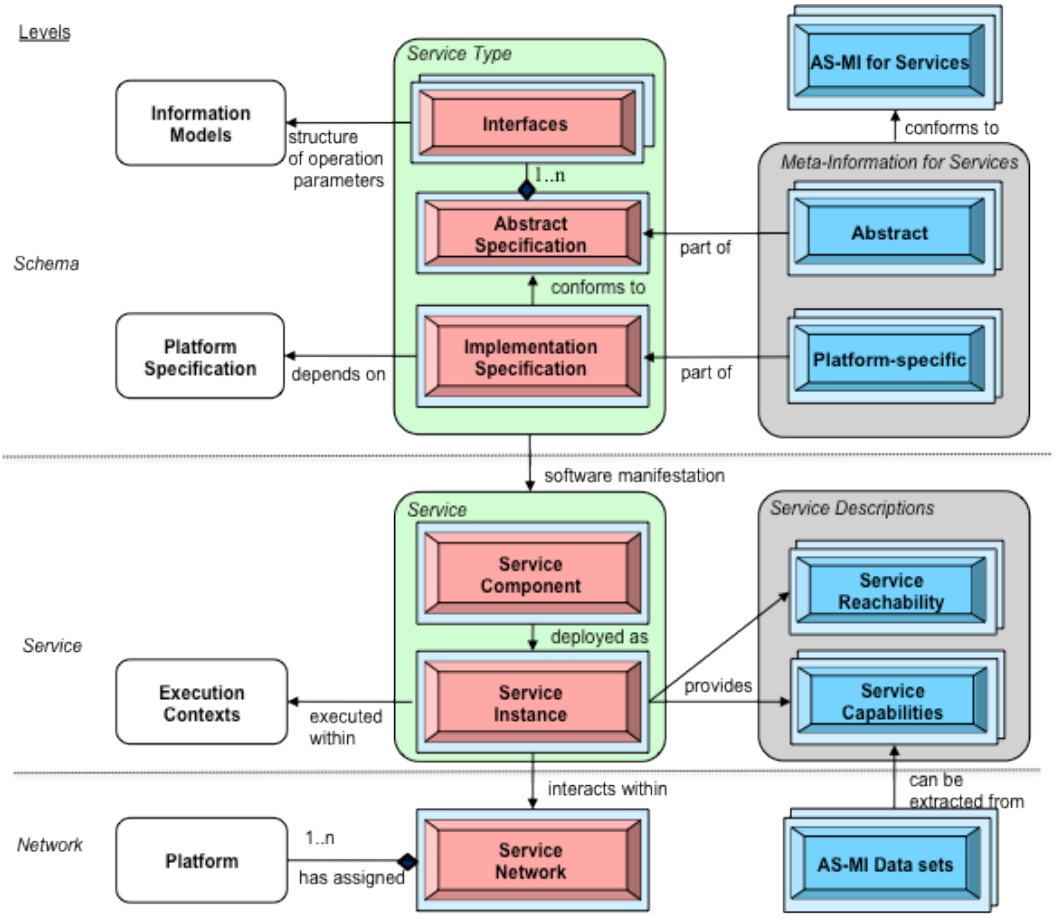


Figure 15: Refined Service Model

In accordance with to ISO/DIS 19119, a Service Type is defined in terms of its Interfaces only (meaning that there is no formal specification of a service type except for its set of interfaces).

An interface also describes the operations and information models related to interactions based on these operations. An operation is a transformation or query that a service may be called to execute, having a name and a list of parameters. The decomposition of service types into interfaces facilitates to reuse interfaces and it is easier to achieve functional interoperability between service consumers and providers of different services, if common interfaces are used.

A service type has two manifestations

- The Abstract Specification that is platform-neutral and typically uses UML diagrams.
- The Implementation Specification that is platform-specific and typically given in an XML format, e.g. WSDL or WADL, containing the description of the service operations and bindings.

The implementation specification must conform to the abstract specification being related by a mapping from the abstract to the implementation specification where

1. There may be several implementation specifications related to one service type.
2. Operations and parameters that are marked on the abstract level as "optional" may be omitted in the implementation specification.
3. The interface types will not be visible in the platform specific level41 but the operations are mapped onto an action model that is part of a so-called Service Interface (according to the OASIS-SOA-RA terminology (for details see Section 7.4.2).

LifeWatch will consider meta-information as part of the abstract respectively implementation specification. The structure of the meta-information within the specifications is determined by the application schemes for meta-information according to purpose. Particular pieces of meta-information may relate to different level. For instance, some meta-information related to "discovery" such as the endpoint of a service is only available for a service instance while other information such as about capabilities will at least be available for the implementation specification. For "orchestration" meta-information about the behaviour in terms of, e.g., message sequence charts may already be part of the abstract specification. The LifeWatch policy will be that meta-information should be provided whenever it is available.

The format of service types, interface types, etc. are determined by meta-classes that are defined in Appendix C. This applies as well to built-in operations of feature types. An example of a complete service specification that applies the rules of the ORCHESTRA Meta-Model is the Catalogue Service presented in C.1 in terms of:

- An informal description in text format,
- A platform neutral specification in terms of UML, and
- A platform specific specification as web services.

On the service level, a Service Component is the software component that implements an implementation specification. The instantiation of a Service Component through its deployment, e.g. on an application server, is called a Service Instance.

All the service instances form the Service Network. The service network may incorporate service instances defined relative to several platforms.

A Platform, describes an architecture specific paradigm e.g., Web Services or Grid based services. The platform determines the static part of the execution context. The Platform Specification is comprised of

i. Interface Language - Formal language used specification of interfaces
ii. Execution Context - Specification of the Execution Context as an agreement between service providers and consumers that contains information about, e.g., preferred protocols, semantics, policies, and other conditions and assumptions that describe how a service can and may be used.
iii. Schema Language - Specification of the schema language for defining Information Models.
iv. Schema Mapping - Specification of how to map platform neutral information model to the schema language used for the particular platform.

41 in that interfaces are just a structuring mechanism on the abstract level.

v. Information Model Constraints - Specification of the constraints on the LifeWatch RM Information Model, especially the constraints on the message format required to accomplish the LifeWatch RM Action model.

A platform shall additionally describe how visibility is achieved between service providers and service consumers. Publishing a description of service capabilities and service reachability contributes to visibility.

A mandatory requirement for LifeWatch Services is that they provide the ServiceCapabilities interface. The ServiceCapabilities interface has one operation getCapabilities that informs the client about the capabilities of a Service Instance. The capabilities describe the interfaces, providing references about the information models used in the interface operations, and the service meta-information. The service meta-information should be accessible according to purpose. Therefore, the capabilities response provides different sections for the included purposes.

An alternative view: Resource Oriented Architecture

The service definition above is based on ISO 19119 that reflects two traditions: that of remote procedure calls (RPC) and information hiding as promoted by object-oriented architectures, which results in the idea that interfaces consists of operations and that data can only be accessed and changed by applying these operations. A second tradition42 (of database origin) lead to a different – more recent - architectural style: Resource Oriented Architectures (ROA). Here the idea is that a "service" offers a set of resources, each of which supports an interface that only consists of operations familiar from http: namely get, put, delete, post. The contrast is obvious: information hiding is abandoned in favour of direct access to resources (data), indirect access to data (resources) via operations are replaced by direct access operations (more details are given in the "Status Report on Infrastructures for Biodiversity Research" (Deliverable 5.1.4), Appendix B where the two architectural styles are compared).

Unfortunately, the two styles result in quite different perceptions of the "real world" leading to different design decisions and there is yet no approach that combine both architectural styles in a uniform framework ranging from the level of abstract specification to concrete implementation. However, there are reasonable rules of thumb how to translate the architectural style as represented by ISO 19119 to the resource-oriented style often tagged with the term RESTful architecture. Hence, this reference model will adhere to the ORCHESTRA RM in following ISO 19119 but indicate in the technology viewpoint (Section 8) and in the Status Report (Deliverable 5.1.4), Appendix B, how the information models and services can be structured in a RESTful way.

6.4 Service description

The OASIS draft proposal "Reference Architecture for Service Oriented Architecture" OASIS-SOA-RA43 rather precisely captures the elements of Service Description that defines the information needed to use, deploy, manage, and otherwise control a service. The LifeWatch Reference Model will base service description on the OASIS proposal combining it with the ORCHESTRA approach.

The elements of the OASIS Service Description are outlined in Figure 16.

The elements of the service description relate to platform-neutral or platform-specific level or to both. A detailed account of these elements as well as allocation to the different levels will be discussed under the

---

42 sometimes referred to as CRUD that is the acronym of the basic operations *create*, *read*, *update*, and *delete* of databases.

43 http://docs.oasis-open.org/soa-rm/soa-ra/v1.0/soa-ra.pdf

Engineering Viewpoint (Section 7.4). The LifeWatch Services Meta-Model reflects the OASIS reference architecture (see Appendix C).

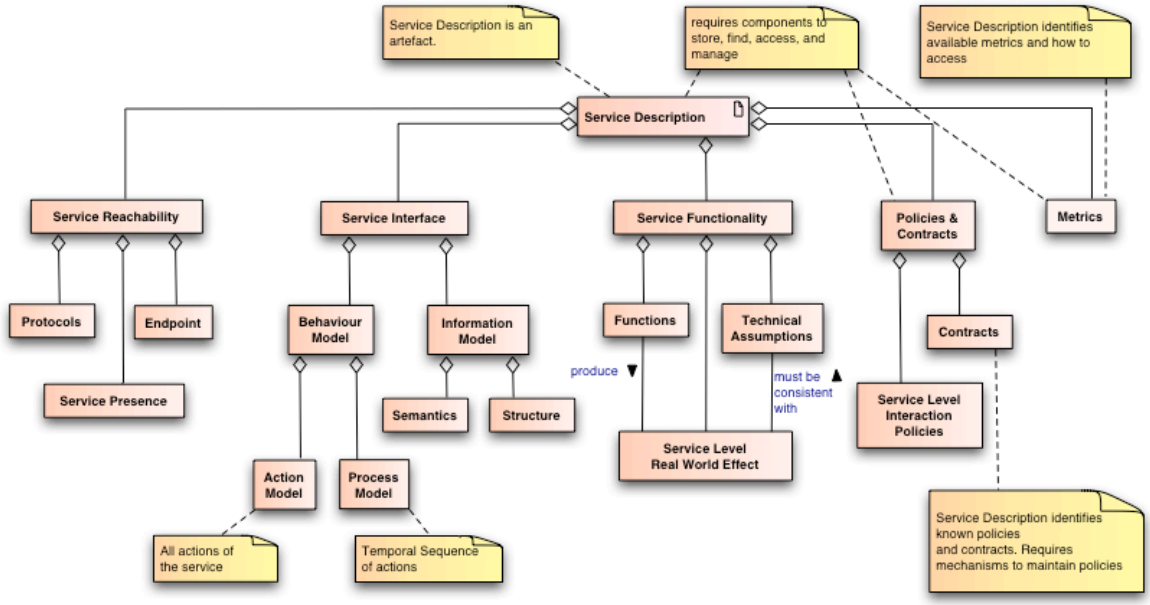


Figure 16: OASIS-SOA-RA class diagram of a service description

6.5 Classification of LifeWatch service types

6.5.1 Network categories

LifeWatch classifies (as ORCHESTRA) service types into service categories according to their role in the service network.

- Basic services are generic services that cover general infrastructural needs of LifeWatch, as opposed to services that emerge from the needs of particular applications or application areas. LifeWatch Basic Services, provide the "backbone" of LifeWatch44.
- Particular application areas may develop services of particular interest for the respective areas. These are referred to as LifeWatch Thematic Services.

Together, Basic Services and Thematic Services make up the Biodiversity Research Capabilities described in the Enterprise viewpoint (section 3.5).

ORCHESTRA classifies service types according to usage rules, which should be adhered to for LifeWatch. Therefore, LifeWatch introduces the term of supporting services for both service categories (basic and thematic services) defining the following categories and relationships:

- Core Basic Services are those LifeWatch basic services, which are indispensable to operate the LifeWatch infrastructure. For them the abstract and implementation specification should be pro-

[44] The concept of Basic Services in LifeWatch is equivalent to the concept of Architectural Services in ORCHESTRA.

vided as well as an excellent documentation of usage. They should satisfy the strictest rules for availability and reliability.

- Supporting Basic Services are basic services that facilitate the operation of the infrastructure. For them the abstract and implementation specification should be provided as well as a good documentation of usage.
- Supporting Thematic Services are generic services, providing common and essential functionalities for a number of thematic areas, like modelling, processing, or taxonomy services. For them the abstract and implementation specification should be provided as well as documentation of usage with examples. They can be considered part of the infrastructure although the providers are attached to a particular domain.
- Specific Thematic Services are those services inherent to a particular domain or application. These may only be specified at implementation level. Services of this category may become supporting thematic services if they are of wider interest than originally thought and if an adequate description is provided.

This classification provides a logical structure in terms of rules for interrelating services. Nevertheless, the category given to a service type must not be rigid and can be changed carefully if required, whenever the changes on the usage relationship to existing services, using the new classified service type does not violate the usage rules given in Figure 17.

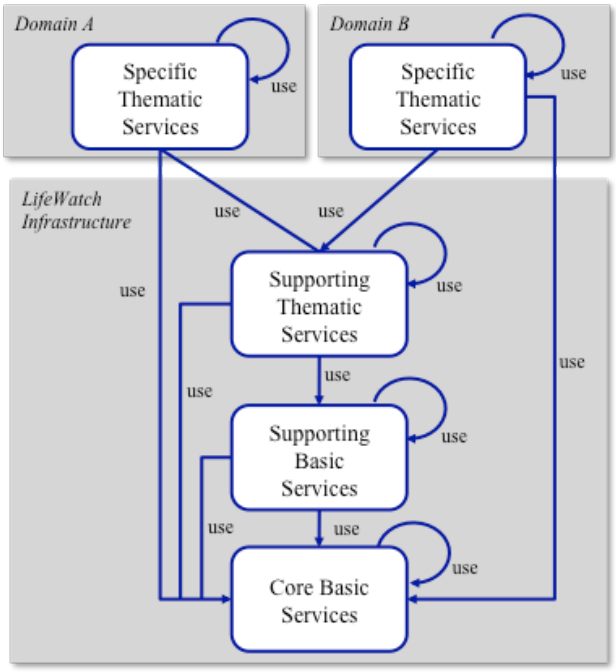


Figure 17: Usage Relationships between Service Categories

6.5.2 Classification of LifeWatch services according network categories

The following list of Services, derived from the ORCHESTRA Reference Model and from the requirements according to the LifeWatch use-cases, gives an overview how to categorize the services. Relevant ORCHESTRA service types for LifeWatch can be categorized according to this diagram as follows:

Service	Short Description
Core Basic Services	
Authentication Service	Verifies genuineness of principals using a set of given credentials
Authorisation Service	Gives a compliance value as response to a given authorisation context

Service	Short Description
Catalogue Service	Supports the ability to publish, query, and retrieve descriptive information for resources data, independent of a specific meta-information standard
Feature Access Service	Allows interoperable read and write access on feature instances available in an Service Network
Document Access Service	Specialization of the Feature Access Service supporting access without manipulation to documents of any type
Name Service.	Encapsulates the implemented name policy for service instances in a service network
User Management Service	Creates and maintain subjects including groups of principals as entities that need authentication.
Service Monitoring Service	Provides an overview about Service Instances currently running within a Service Network
Supporting Basic Services	
Portrayal service (Map and Diagram Service)	Visualizes, symbolizes, and enables geographic clients to interactively visualise geographic and statistic data by providing a graphical representation of the data
Coordinate Operation Service	Changes coordinates from feature locations from one coordinate reference system into another
Schema Mapping Service	Mapping of features into a target schema through the transformation of each data instance from one data structure into another one preserving the original meaning
Thesaurus Access Service	Read and write access to a (multi-lingual) thesaurus for the vocabulary used on a service network
Ontology Access Service	Read access to the specification of a logical ontology, export or import of complete specifications into an ontology store.
Format Conversion Service	Allows the conversion of data given in one format to the corresponding data given in another format
Gazetteer Service	Allows a user to relate a geographic location instance identified by geographic names with an instance identified by coordinates
Service Chain Access Service	Supports the creation of an executable service instance based on an explicit description of a service chain The execution of the chain is outside scope of the service
Calendar Service	Transformation, comparison and arithmetical operations on data/time functions
Annotation Service	Automatically generation of specific meta-information from various sources and relation with semantic descriptions
Provenance Service	Specialization of the Feature Access Service for accessing provenance data generated through annotation services, among others
Communication Service	Harmonized access to direct user-to-user communication means based on multi-media technology and data-exchange between users (like chat, teleconference, SMS)
Processing Service	Common interface for services offering processing operations by initiating the calculation and managing the outputs to be returned to the client.
Workflow Enactment Service	Specialization of processing service Allows the execution and monitoring of a workflow or a service chain
Compression Service	Performs data compression

Service	Short Description
Notification Service	Allows to send messages to a client, which previously has been registered to listen for certain events
Supporting Thematic Services	
Sensor Access Service	Specialization of the Feature Access Service for accessing sensor data, configuring a sensor and publishing sensor data
Modelling Services	Specialization of processing services, allows the user to discover, specify input for, and control execution of a variety of models (e.g. for simulation)
Project Management Support Service	Supports the planning and performance of operations (projects) in a cooperative distributed environment
Reporting Service	Creates reports using actual information from other services according to a template (wrapper interface for existing products)
Taxonomy Access Service	Specialization of Feature Access Service Allows to read (and write) taxonomic information
Geolinking Service	Allows establishing a virtual join between data having a spatial location and data without spatial location but referring to the same feature through common properties
Specific Thematic Service (examples)	
Sensor Planning Service	Interface to collection assets and support systems around assets
Geocoder Service	Allows adding geographic information to address data
Generalisation Service	Allows to create spatial, temporal and other generalisation of features according to a given hierarchy
Interpolation Service	Allow to interpolate spatial locations
Occurrence Distribution Service	Allows creating distribution maps from a particular specie or specie group

Note: The list of Basic Services shall be incrementally updated during the preparatory and part of the construction phase.

The basic services and the support thematic services represent the basic Biodiversity Research Capabilities (Section3.5). The specific thematic services are extensions that reflect particular areas of biodiversity research.

Abstract and implementation specifications for existing Basic Services45 shall be used where possible in order to reduce the effort required for development of new LifeWatch specifications, and to improve interoperability with external systems. The Open Grid Services Architecture (OGSA), for example, provides services for Authentication and User Management, in addition to other useful services such as Job Execution, Data Access, and Accounting. Additional Basic Services will be derived from the biodiversity domain and added to this set.

The examples listed as specific thematic services were derived from requirements of preliminary LifeWatch show cases (in "Biodiversity Themes and Research Capabilities", Deliverable 5.1.2). However, the examples listed above may be relevant for several thematic areas, hence may become supporting thematic services. For the construction strategy, these specific thematic services may be prioritized for subsequent implementation. Therefore, they are described in more detail below. All these services should be specified as OGC conform Web Processing Services (WPS).

[45] e.g. defined as ORCHESTRA Architectural Services and available at http://www.eu-orchestra.org, last access August 2009

Sensor Planning Service

Monitoring applications in biodiversity / environmental domains require service interfaces for sensors and sensing processes (e.g. data fusion and visualization)46. Emphasis has been to date on how to collect and process data rather than to consider the kind of data that should be collected in order to meet information quality requirements for different tasks47 and the needs of end-users (of various kinds)48. The ability to dynamically configure sensor data collection to meet the information needs of scientists is foreseen as a valuable mechanism to support increased and targeted data generation.

The Sensor Planning Service will include a set of service capabilities that encompass:

1. Capture of user needs (in terms of their information requirements),
2. Data collection,
3. Information processing, and
4. Information delivery.

These four kinds of service form a process loop, where the output of each service feeds into the next: users' needs drive data collection, the data drives information processing, derived information is delivered to users in a suitable form to aid their comprehension, often leading to needs for further information. These service capability and process definitions demands for exploratory prototyping.

Geocoder Service

Geocoding refers to the transformation of textual location information such as street addresses, country names, landmarks, or even phone numbers into a normalized location description (feature) with geographic geometries described in terms of coordinates based on a coordinate reference system. Gazetteer services are a more specialized form of geocoders, focusing on searching feature hierarchies and discovering feature types. Furthermore, geocoder services often offer a batch-modus for processing many entries.

Implementations of geocoder services rely on different algorithms and vocabularies needed for normalizing the geocoding entries, e.g. addresses or for interpolating house numbers along a street geometry. The quality of the transformation depends also on the geographic data used in the algorithms. An OGC Geocoder Service draft candidate implementation specification was submitted 2001, but has been deprecated by the consortium.

The LifeWatch community may need geocoder services for specific purposes, as for georeferencing documents (e.g. publications with an address) and for adding coordinates to derived metadata from datasets in standard formats like DwC or ABCD at a single blow.

Generalisation Service

Generalisation of spatial data reduces the level of detail of geographic data. It comes in two flavours:

- Information contained in spatial features is aggregates along a hierarchical system. It allows users to define at which spatial scale the data should be provided, using predefined scale levels as, e.g. exact locations, regions, countries, etc. or using spatial grids with different sizes.

---

46 Open Geospatial Consortium: Sensor Web Enablement, http://www.opengeospatial.org/projects/groups/sensorweb

47 Rowaihy H, Johnson MP, Pizzocaro D, BarNoy A, Kaplan L, La Porta T, Preece A,. Detection and Localization Sensor Assignment With Exact and Fuzzy Locations. in Proc 5th IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS'09). 2009: Springer

48 Mohyuddin, Gray WA, and et al. Development of Patient Centric Virtual Organisations (PCVOs) in Clinical Environment for Patient Information Management in Medinfo Proc 12th world Congress on Health (Medical) Informatics. 2007. Brisbane, Australia

- Information contained in feature geometries is decreased by applying geometry generalisation algorithms. These reduce the number of points in a polyline without loosing too much information. This approach helps to reduce the amount of transferred data between systems.

For LifeWatch the first usage will be necessary to provide different aggregations levels on one hand and, on the other hand, may be needed for constraining the access to very detailed information, e.g. for the exact location of endangered species.

Data Construction Service

Biodiversity experiments based on observation samples often need to create new data based on existing data sets in order to analyse a spatial region for which not all areas are covered by species survey data. For this purpose, mathematical operations like interpolation and extrapolations are applied, where new data points are constructed within and accordingly outside the set of known species locations. Those methods are very popular for creating valuation maps, for bird strike monitoring, for calculating scenarios, etc. Services offering specific construction methods, e.g. interpolation service, may be provided as specialisations of data construction services.

Species Occurrence Service

The Species Occurrence Service provides an opaque workflow for querying species occurrence within a spatial or temporal extent. Given occurrence data, the service should provide means for identifying the taxon and plot the aggregation of occurrences for the given spatial level and temporal scope in a map or alternatively providing vector/grid information in a standardized format (e.g. KML or GML, netCDF). This service can be used within different workflows and can also support the creation of INSPIRE services for species distribution (Annex III). The GBIF Web Service "Occurrence record data service" is an example implementation providing access to XML-formatted (mainly as DwC or KML) GBIF data. LifeWatch may provide a LifeWatch conform access to this and similar GBIF services.

Species Distribution Service

The Species Distribution Service provides an opaque workflow covering the tasks of applying distribution models to specimen occurrence data using abiotic and biotic factors. The service should provide maps or vector/grid information allowing the requestor to choose model, factors to be considered, and data source. Species Distribution Services will probably use a Species Occurrence Service. These service types may be the basis for providing INSPIRE services for species distribution (Annex III).

6.5.3 Taxonomic categories

Services can also be classified according to their functionality by grouping services having a similar behaviour or goal. ISO 19119:2005 and the OGC defined 5 main Service Taxonomies, namely

- Human interaction services: Management of user interfaces
- Model/Information services: Management of meta-information, schemas and datasets
- Workflow/Task management services: Support of specific sequence of activities
- Processing services: Perform computations
- System management services; Management of system components, applications and networks

A classification of the services named above according to the ISO taxonomy is given in C.2.

6.5.4 INSPIRE network service types

The INSPIRE[49] Directive specifies a set of common Implementing Rules to ensure, that the Spatial Data Infrastructures (SDIs) of the member states are compatible and usable in a transboundary context. The technical architecture of SDIs is described through Network Services. The INSPIRE Network Services Architecture[50] defines the INSPIRE Service Types as mandated by the directive in terms of their interoperable interfaces. For each service, type a detailed definition for their implementation is given in an Implementing Rules document. Figure 18 shows an overview of the INSPIRE service types (Registry Service, Discovery Service, View Service, Download Service, Transformation Service and Invoke Spatial Data Services (SD) Service). Applications and geo-portals can access the different services via the INSPIRE Service Bus, which is defined by the standard interfaces of the INSPIRE network services. In most of the cases the services are protected through a Geo Right Management (GeoRM) layer, which control the access to services under operator –defined conditions, business models and policies.

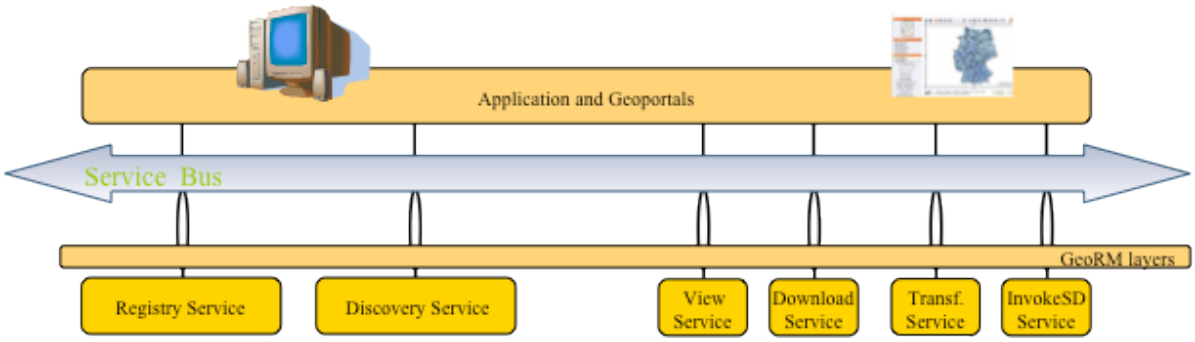


Figure 18: INSPIRE Service Types[51]

LifeWatch shall comply with the Implementing Rules for each service type by providing conformant interfaces for each INSPIRE service type. For example, the service type specification for Catalogue Service will provide an interface for the querying of meta-information about discoverable resources, according to the Implementing Rules for DiscoveryService[52]. The interface can be based on the ORCHESTRA CatalogueSearchInterface and can extend the INSPIRE normative rules by defining a LifeWatch profile of the information models specified by INSPIRE, for example, for the exchange of Capabilities. On the specification of the interfaces a reference to the corresponding INSPIRE Implementing Rule should be given.

6.5.5 Concluding remarks

ORCHESTRA provides the abstract specification of a number of service types together with their implementation specifications for the web platform. With the development of the LifeWatch information and meta-information models, it remains to be seen what modifications are needed for these services and whether additional basic services are to be specified. A need of such services is foreseen in thematic application areas such as taxonomy or collections. Identification and specification of such services will ev-

---

[49] INSPIRE is the European Union directive to create an European spatial data infrastructure (http://inspire.jrc.ec.europa.eu/)

[50] See INSPIRE Network Services Architecture, Draft, 19-07-2008, Identifier D3_5_INSPIRE_NS_Architecture_v3-0.pdf, available at http://inspire.jrc.ec.europa.eu/ (August 2009)

[51] Source: INSPIRE Networl Service Architecture Version 3.0, available under http://inspire.jrc.ec.europa.eu/reports/ImplementingRules/network/D3_5_INSPIRE_NS_Architecture_v3-0.pdf, November 2009

[52] The current version of the technical guidance for INSPIRE Discovery Services is available under http://inspire.jrc.ec.europa.eu/documents/Network_Services/Technical%20Guidance%20Discovery%20Services%20v2.0.pdf (August 2009)

olve in collaboration with the application specialists as part of the product management strategy for LifeWatch.

A further area to be investigated already in the preparatory phase concerns the implementation specification for the Grid platform, i.e. as grid services. The issue here is which OGSA services have to be adopted into the platform neutral specification – services for managing VOs, certainly.

6.6 Service chaining

The LifeWatch infrastructure intends to "provide evolving technologies for data generation, data processing, data integration, and the creation of virtual laboratory environments encouraging analysis, modelling, and experimentation to support scientific research and decision making". This ambition implies that LifeWatch, besides providing capabilities for discovering and accessing biodiversity data, should support the following:

- The scientific method of constructing and testing hypothesis via repeatable experiments
- The communication of results of a scientific method
- The integration of tools involved in a specific task
- Value adding mechanisms: existing information or capabilities should be enhanced by supplying additional information or by adding new capabilities, which can be provided from other existing structures, in order to deliver new results, which comprise benefits, or value, to the consumer
- The easy creation of applications by combining services.

These requirements have in common the need for capabilities to combine existing components in a persistent and reusable way. This is a matter of specifying how the components can interact with each other. Since the LifeWatch basic components are services, the answer is to support service chaining, which is referred to in LifeWatch as Workflow. This section introduces the concept of "Workflow" as LifeWatch will use it. The specification refers to the Workflow Reference Model (TC00-1003 V. 1.1. 1995), specified by the Workflow Management Coalition (WfMC)[53], since it provides a generic framework to support the development of workflow management systems identifying their characteristics, terminology, and components. Although the reference model is more than 10 years old, it is still relevant today (see statement of the WfMC Technical Commission).[54] Some suggestions for the implementation specification will be also given regarding the current state-of-the-art of workflow management systems and standards.

6.6.1 Core workflow concepts

Workflow definitions

From the research perspective, a Workflow is the "process of combining data and processes into a configurable, structured set of steps that implement semi-automated computational solutions of a scientific problem" [Wikipedia, Kepler].

The Workflow Management Coalition (WfMC) defines workflow as "The automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules". In other words, a workflow consists of all the steps that should be executed in order to deliver an output or to achieve a goal. These steps (tasks) can have a variety of complexity and are usually connected in a non-linear way, forming a directed acyclic graph (DAG). A Workflow Management System defines, manages, and executes workflows through the

[53] http://www.wfmc.org/

[54] The Workflow Reference Model - 10 years on: http://www.futstrat.com/books/downloads/Ref_Model_10_years_on_Hollingsworth.pdf

execution of software that is driven by a computer representation of the workflow logic. The description of a workflow includes the definition of different tasks, their interconnection structure, and their dependencies and relative order. This description of the operational aspects of a workflow can be expressed in textual (e.g. XML) or graphical form (e.g. as a graph in Business Process Modelling Notation [BPMN] or Petri nets).

In the context of Grids, a workflow is postulated as "The automation of the processes, which involves the orchestration of a set of Grid services, agents and actors that must be combined together to solve a problem or to define a new service".

In the context of OGC, workflows are produced through "service chaining", which can be performed in a number of ways:

- Orchestration of a service chain including one or more Web Processing Services (WPS) using a BPEL engine.
- A WPS process can be designed to call a sequence of web services including other WPS processes, thus acting as the service-chaining engine.
- Simple service chains can be encoded as part of the execute query. Such cascading service chains can be executed even via the GET interface.

Workflow components

For LifeWatch a Workflow will be defined as: an identifiable and thus reusable sequence of services, including the related resources and agents, where:

- Services are the basic functional components of the infrastructure, defined by interfaces,
- Resources are all elements to be used by the service such as information, information models (see Section1), or other services, which are used internally and are not explicitly visible for the workflow and,
- Agents are the actors involved in the workflow having particular roles and responsibilities (e.g. observer, creator, consumer, etc.) and may be a human or another service.

The principal components and their interactions are depicted in Figure 19.

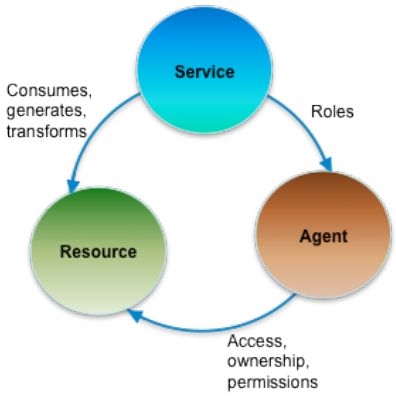


Figure 19: Workflow components

The WfRM defines two major phases for workflows: (1) a build phase, where the workflow is defined in terms of a textual or graphical language, and (2) a run phase where the workflow is enacted according to its definition by a workflow execution (enactment) component or a workflow engine. The capabilities required for these phases leads to a functional decomposition of workflow management into the following elements:

- Workflow composition: user environments, usually graphical, where the user can define a workflow (phase 1)
- Workflow definition: representation languages that are used to express workflows (phase 1)
- Workflow compilation: Translation or compilation of a workflow so that it could be enacted (phase 2)
- Workflow enactment: execution of a workflow and runtime support (phase 2)

To support the ability to work with workflows, LifeWatch will provide separated workflow management capabilities that can be used independently in support of each of these elements.

The nature of workflow can be very varied. Depending on the duration of execution, results can be delivered quasi immediately or a notification mechanism may be needed to inform the user how to retrieve the results, when the workflow finished. The number of entities (actors, services and resources) and the way they are organised involved can also vary significantly. Different workflow categories can be derived according to the different aspects.

Workflow patterns

This aspect reflects options for the allocation of workflows to components, according to different priorities for different applications. According to the design pattern for service chaining defined by OGC, LifeWatch introduces three types of workflow representing different infrastructure components holding the workflow definition and users' perspectives on workflow [cf. OpenGIS Service architecture]:

- User-defined (transparent) chaining: The human supervises and controls the workflow execution. The user knows how services can be combined, is able to discover available services matching his/her goals, guarantees that inputs and outputs of the services chained are compatible, provide sufficient resources to run the services, and controls execution. Calling services sequentially using a terminal with line wise commands is, for instance, an example of transparent chaining. A portal with predetermined interactions by a user that result in calling different services sequentially, where the output of one service determines which services may be called next, may be considered as a transparent chaining (though the user may not be aware of services being called).This pattern is also known as client-coordinated service chaining.
- Workflow-managed (translucent) chaining: The user calls a workflow management service (Enactment service) that controls execution of the chain but is aware of the structure of the workflow and the individual services chained. A key distinction to transparent chaining is that al of the workflow is defined before execution. While executing, the user's role is mostly one of observing the progression of execution of the individual services but the user may be asked to interact, for instance, in order to provide specific parameters. The enactment service handles the details of execution, for instance, of distributed computing. Depending on intermediate results the user may interfere, for instance, to abort execution due to lack of convergence of computation. The typical scenario is that the user has edited a workflow using a Workflow Editor Service or that a user retrieves a workflow from a repository and then calls an Enactment Service.
- Opaque chaining: The workflow has been deployed as a service, which performs an aggregation and management of the related services. The user has no awareness of the individual services and relatively little control over the workflow. This type is often hidden behind portal functionality or complex processing services. A portal with predetermined interactions by a user that result in calling different services sequentially, where the output of one service determines which services may be called next, may be considered as a opaque chaining since the user is not aware of the underlying workflow. This pattern is also known as aggregate service or static chaining.

Figure 20 shows collaboration diagrams for the three patterns.

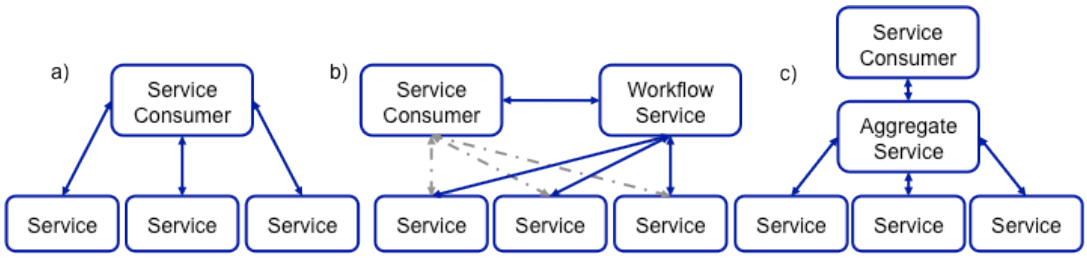


Figure 20: Workflow patterns: a) transparent workflow; b) translucent workflow; c) opaque workflow

Execution models

- In data-driven workflows, information is streamed from one agent to the next. An activity is initiated by the arrival of data. Often, data-flow based workflows have no explicit control instance for the whole process.
- In control-driven workflows, the connections between agents represent transfer of control from one task to another. This typically includes control-structures such as sequences, conditionals, and iterations.
- Hybrid models combine data-driven workflows with discrete control elements.

Workflow representation

This aspect is concerned with the structure of workflows. The following structure of increasing complexity:

- In a linear sequence: simple sequence of tasks to be performed in a specified linear order;
- As an acyclic graph: some workflow tasks depend upon the completion of several other tasks which may be executed concurrently;
- As a cyclic graph: the cycles represent some form of explicit or implicit loop or iteration control mechanism. Services are "connected" with each other and communicate via messages or sequence of transactions; and,
- As a workflow hierarchy: is used to model very complex workflows, which can be separated into individual graphs. The workflow hierarchy represents a workflow of workflows, where the control instance is not necessarily is aware of the sequences defined within a workflow part.

6.6.2 Semantic implications of workflows

The use of workflows leads to a higher complexity level of semantic considerations. Besides using semantic annotations for resource discovery and proofing whether a resource is suitable for a particular purpose, workflow consumers need enhancement to chain two services together, not just by syntactically compatibility but also semantically. Furthermore, it is important to determine if workflow results are valid for the problem to be solved and if they are comprehensible and reproducible. An approach for semantic enhancement of workflows is the introduction of Task Ontologies, which may be abstract from a specific domain or organisational context and more attached to the necessary knowledge to solve the specific problem or task.

Validity

It is assumed that human users will determine semantic validity of the results of a service chain. Several factors to consider in the semantic evaluation of a chain result are listed in ISO 19119:

- Appropriateness of starting data: are the based datasets suited to the subsequent processing? For example, accuracy, and resolution of the data, thematic values are relevant.

- Affect of services on data: how do the individual services affect the data, e.g., error sources and propagation.
- Sequence of the services: how does the order of the chain affect the results? For example, should a spatial operation, e.g., ortho-rectification, be performed before or after a thematic operation, e.g., re-sampling the attribute values?

The evaluations depend upon the user's understanding of services and their combinations.

Mediation

The construction of workflows implies that the services used are aligned. Mediation is an interoperability mechanism outside the service components providing semantic matching of service capabilities. Semantic matching is needed when two services to be chained have incompatible exchange information models or when the service visibility elements should be matched to a particular request. For example, service interfaces of a service instance providing the desired functionality may not conform a particular workflow definition either due to an update or a new version of the interfaces or because they are specified using a different vocabulary. Mediation services may be manually or automatically inserted into a workflow to align the input and output of closely related data or operation name, when necessary. They are also known as adaptor, façade, shim services, or simple shims.

Relevant elements for mediation are:

- A semantic enhanced description of services to allow semantic matching[55]
- A matching vocabulary to describe relationships between services and matching degrees[56]
- Definition and specification of different mediator service types to solve the matching according to different contexts. The context may be determined either through particular information models or through considering different matching strategies, like translation, dereferenciation, comparison, parsing, etc[57].

Provenance

Just as the results of a conventional laboratory experiment are captured by a laboratory journal and thus are reproducible, computational "in-silico" experiments and runs of scientific workflows should be captured and indicate which specific data products and tools have been used to create a derived data product. This implies, beside capturing metadata, that all the applied steps, parameter settings used, and important intermediate data products need to be recorded in a format suitable for archival and exchange as well as for inspection by humans.

At present, there are many open questions related to generation of provenance data for workflows, the most important ones being about the proper level of granularity and, related, the model used for provenance. The Open Provenance Model[58] attempts to define a standard but whether this standard will be ad-

---

[55] See Massimo Paolucci, Takahiro Kawamura, Terry R. Payne, and Katia Sycara. Semantic Matching of Web Services Capabilities, 2002. International Semantic Web Conference (ISWC) and Duncan Hull, Robert Stevens, and Phillip Lord. Describing Web Services for user-oriented retrieval. In W3C Workshop on Frameworks for Semantics in Web Services, Digital Enterprise Research Institute (DERI), Innsbruck, Austria. June 9-10, 2005

[56] See Duncan Hull (2008) Semantic Matching of Bioinformatic Web Services, A thesis submitted to the University of Manchester for the degree of Doctor of Philosophy in the Faculty of Engineering and Physical Sciences, School of Computer Science.

[57] See Duncan Hull, Robert Stevens, Phillip Lord, Chris Wroe and Carole Goble. Treating "shimantic web" syndrome with ontologies. In First Advanced Knowledge Technologies workshop on Semantic Web Services (AKT-SWS04) KMi, The Open University, Milton Keynes, UK. 2004-12-08. (See Workshop proceedings CEUR-WS.org (issn:1613-0073) Volume 122 - AKT-SWS04)

[58] http://eprints.ecs.soton.ac.uk/16148/1/opm-v1.01.pdf

opted is uncertain at the present stage of development. Factual situation is that some workflow systems such as, e.g., Taverna or Kepler come along with a provenance model of their own using proprietary formats for provenance (as they do for workflow specification).

Blank page

7 Engineering viewpoint

7.1 Introduction

According to the ORCHESTRA Reference Model, the Engineering and Technology viewpoints are not in the scope of the ORCHESTRA Architecture. Instead, they are combined in one or more dedicated specification step that map the ORCHESTRA Architecture (the Information and Service Viewpoint specification) to a specific service platform, leading to a platform-specific ORCHESTRA Implementation Specification. Hence, the ORCHESTRA Reference Model only provides guidelines, requirements, and rules for defining ORCHESTRA Implementation Specifications, addressing the specification of a platform in case of the Technology Viewpoint, and addressing the mapping to a chosen platform in case of the Engineering Viewpoint.

While in principle proceeding being in accordance with ORCHESTRA, the LifeWatch Reference Model extends the Engineering Viewpoint by providing elements of a reference architecture relying on the OASIS draft proposal "Reference Architecture for Service Oriented Architecture" OASIS-SOA-RA59. This Reference Architecture makes a number of key assumptions that shared by LifeWatch, namely that

- Resources are distributed across ownership boundaries.
- People and systems interact with each other across ownership boundaries.
- Security, management, and governance are also distributed across ownership boundaries.
- Interaction between people and systems is primarily through the exchange of messages wit appropriate reliability with regard to the intended uses and purposes.

The OASIS Reference Architecture specifies architectural elements and relations between these in terms of three different perspectives60

- Business perspective – intends to capture what the system means for users describing the role and responsibilities of stakeholders whose concern is (i) to use the system in a safe and efficient way and/or (ii) to organize the relationship of users and organisations as decision makers. The architectural elements of the business perspective relate to the Enterprise Viewpoint in that the form a basis for modelling the people involved, their goals and activities as well as the relevant interactions including modelling of governance and social structures (as embodied, for instance, by legal or quasi-legal frameworks).
- Construction perspective – focuses in the infrastructural elements that are needed to support the construction of a system. In particular, it deals with service description, service visibility, service interaction, and service policies and contracts. In contrast, to Section 6, the focus is on the architectural elements needed for (inter-) operability of services and service nets rather than on the concepts and particular service types needed.
- Ownership perspective – addresses the issues of owning and managing the system. It is concerned with how (evolving) systems are managed effectively, how decisions are taken and made known to the required endpoints, how to ensure that people may use the system effectively across organisational borders, and how to avoid corruption by malicious users. The ownership perspective addresses governance, security, and management and will be treated in Section 7.5.

The present version of the LifeWatch Reference Model will not consider the business perspective. Nevertheless, capturing the business perspective(s) is an important aspect of the LifeWatch project that should be filled in some future LifeWatch document.

---

59 http://docs.oasis-open.org/soa-rm/soa-ra/v1.0/soa-ra.pdf

60 Originally, SOA-RA speaks of "viewpoints" as well. This document shall speak of "perspective" in order to avoid confusion with the ODP-RM terminology.

Following the ORCHESTRA Reference Model, the engineering viewpoint will state guidelines, requirements, recommendations, and rules in terms of policies. According to OASIS-SOA-RM, "a policy as a representation of some constraint or condition on the use, deployment, or description of an owned entity as defined by any participant". At the present stage, policies will be considered as a first recommendation for the construction phase. Further development and consolidation of the architectural elements and the related policies shall go hand in hand with the construction phase of LifeWatch to become mandatory. Policies must be considered as an inherent part of services and service networks and as such be handled like other resources within the LifeWatch ICT infrastructure (see Section 7.4.4).

The presentation of engineering viewpoint starts with a section on "Engineering Guidelines" stated in terms of policies. These policies reflect standards and good practice. Then service networks (Section 7.3) and service descriptions (Section 7.4) as well as the ownership perspective (Section 7.5) are discussed reflecting the concepts of OASIS-SOA-RA and ORCHESTRA-RM. A section on LifeWatch's technical capabilities (Section 7.6) considers specific technical aspects to be provided by the LifeWatch ICT infrastructure. One should note that the LifeWatch Reference Model takes a more holistic view of the Engineering Viewpoint extending the viewpoint beyond being concerned with the infrastructure required to support system distribution as specified by ISO/IEC 10746.

7.2 Engineering guidelines

The engineering of the LifeWatch ICT Infrastructure shall take into account general guidelines and recommendations for the design of service oriented architecture and, more generally, distributed system. These include:

Simple Service Architecture

ISO 19119:2005 lists a set of guidelines to be fulfilled by systems supporting the combination of services in a dependent series to achieve larger tasks (service chaining) referred to as "Simple Service Architectures":

- Message-operations: Operations should be modelled as messages, consisting of a request and response. Parameters should be transferred in a uniform manner independent of content. Request response interactions should follow synchronous or asynchronous message exchange patterns.
- Separation of control and data: A client should have the option of receiving just the status of a service. The data should be accessible separately.
- Stateful vs. stateless service: services should be preferably stateless. Stateful services operations should trigger transitions between the service states.
- Known service type: The service type and its capabilities of a service instance must be accessible to consumer, e.g. through the provision of specification documents. Service type specifications should be available for the consumers.
- Adequate hardware: Hosting issues should be transparent to the user.

Distribution transparencies

Distribution transparencies hide complex distribution issues from a service network. ISO 19119:2005 identifies the following distribution transparencies:

- Location transparency hides where object resides.
- Replication transparency hides the fact that an object or its state may be replicated and that replicas reside at different locations.
- Failure transparency hides failure and possible recovery of objects.
- Migration transparency hides from an object the ability of a system to change that object's location.

- Persistence transparency hides the actual activation and deactivation of objects from a persistent store, and the actual storage mechanisms and representation format used.
- Transaction transparency hides the coordination of activities between objects to achieve consistency at a higher level.
- Resource transparency masks passivation and reactivation of resources.
- Federation transparency masks interworking across multiple administrations.
- Group transparency masks the use of a group of objects to provide an interface.
- Security transparency hides the mechanisms that are being used for Authentication and authorization.

Further distributions transparencies to consider are

- Access Transparency hides differences in data representation and invocation mechanisms
- Concurrency Transparency Clients hides the effect of concurrent access to server interfaces (on objects in the distributed system)
- Relocation Transparency hides from a client the ability of a system to change the location of an object to which the client is bound.

W3C recommendations

The LifeWatch Infrastructure should comply with the bases and principles of the W3C Recommendation "Architecture of the World Wide Web"61, which defines the WWW as an information space in which items of interests are called resources and people or software acting on this information space are called web agents. For the latter, LifeWatch will use the term participants following the usage of OASIS-SOA-RA.

- Identification: Resources are identified through global identifiers called Uniform Resource Identifiers (URI).
- Interaction: Web agents communicate using standardized protocol that enable the interaction through the exchange of message, adhering to a defined syntax and semantic.
- Formats: The interaction protocols place limits on the formats of representation data and metadata that can be transmitted.
- Orthogonality: Orthogonal abstract benefit from orthogonal specification, which may evolve independently (increases the flexibility and robustness of the WWW.
- Extensivity: The technology used should promote evolution without sacrificing interoperability. A method to achieve extensivity is the usage of orthogonal technologies to the architectural framework, for instance URI schemes, or the use of XML-based grammars.
- Error recovery: Agents, which not repair an error condition, but continue processing by addressing the fact that the error has occurred, should do it in a way that is evident to users.
- Protocol-based Interoperability: Protocols are interfaces that specify syntax, semantic, and sequencing constraints of the interchanged messages. The protocol-based design is also known as "contract first" design and has the effect, that the technology shared among agents often last longer than the agents themselves.

Some of these W3C principles rephrase the guidelines of ISO 19119. However, some additional recommendations can be derived.

LifeWatch policy: (i) LifeWatch Service Architecture should comply with the guidelines of ISO 19119 and the recommendations of W3C as applicable.

---

[61] Jacobs, Ian; Walsh, Norman (2004): Architecture of the World Wide Web, Volume One, W3C Recommendation 15 December 2004, available at http://www.w3.org/TR/2004/REC-webarch-20041215/

(ii) From these, explicit policies shall be derived with regard to the particular resources and participants involved.

7.3 Service networks

7.3.1 Basics

A LifeWatch Service Network consists of a set of Service Instances connected through a communication network. A Platform Specification determines all the components of a service network.

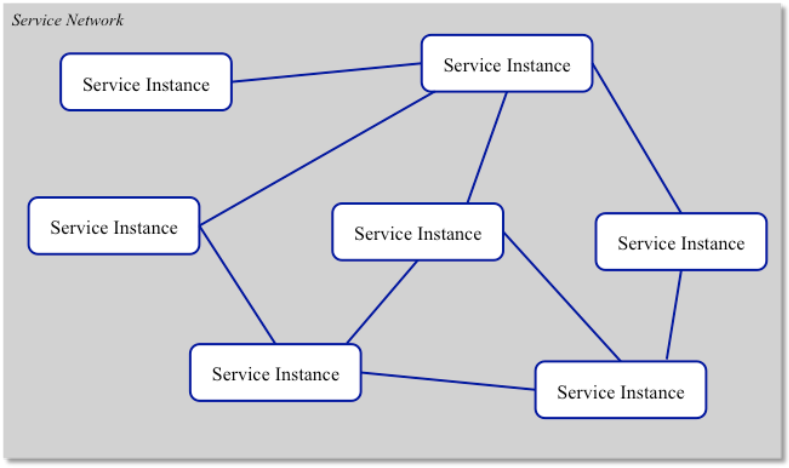


Figure 21: A network of service instants

A Service Instance is a deployment of a Service Component as a software implementation of a Service Implementation Specification that executes on a piece of hardware (often referred to as an Application Server). Service Instances interact in the Service Network. In order to do so, a Service Instance needs to expose a description of itself including information about reachability, functionality, etc.

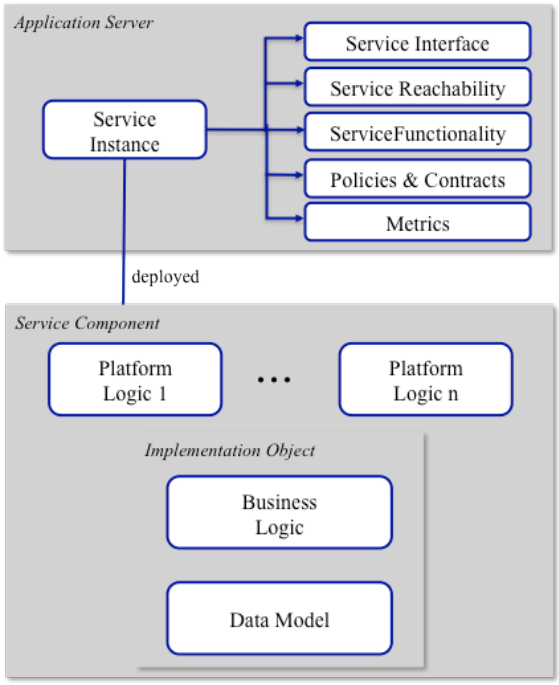


Figure 22: Service Instance and Service Component

A Service Component may be thought of as to consist of an Implementation Object and one or more Platform Logics. The Implementation Object represents the software components behind the service. Usually it comes in two tiers, a Data Model and the Business Logic. Platform Logic is a façade (as design pattern) of the Service Implementation Object (given as, e.g., a piece of Java code) with regard to a specific implementation platform (for instance using SOAP for transport protocol and message format). A Service Component may support more than one Platform Logics. The successful Amazon S3 service, for instance, seems to be organised in this way offering SOAP and REST interfaces, with messages being automatically dispatched according to message format.

Separation between Platform Logic and Service Implementation Object supports reuse. The Implementation Object should be build according to the platform-neutral specification of a service while the Platform Logic provides the software components needed to fulfil the requirements of the platform-dependant service specification.

LifeWatch Policy: *(i) Service Components should separate between Platform Logic and Implementation Object.*

(ii) The Implementation Object should conform to the platform-neutral specification of the service.

7.3.2 Platform

According to ORCHESTRA, the communication network is platform-specific as is the Service Implementation Specifications. However, some requirements or assumptions can be stated that apply for services and the communication layer that are independent of particular platform architecture. These have to do with

- The interface between services and the communication network referred to as Service Interface (this should not be confused with "Interfaces" that are structuring mechanisms for exposing the functionality of a service).
- The interaction of services within the communication network.
- The visibility of services in the network.
- Policies and contracts that apply to services in a network.

The Engineering Viewpoint identifies and models certain aspects that shall be common to all LifeWatch Service Networks as a means to enhance interoperability between service networks based on different platforms. OASIS-SOA-RA serves as a blueprint. Which features and entities are required will be specified in terms of policies.

A Service Network captures the conditions under which interaction can occur, for instance, the message exchange patterns and protocols that are supported, its core services, the (business) processes these are engaged in, architectural patterns used, and the policies and contracts that apply. The LifeWatch Core Services have been specified in Section 6.5.3. Architectural patterns, core business processes, and policies related to the core services will be discussed below (Section 7.6).

7.3.3 Message exchange

A basic, generally shared, assumption about the communication layer of service-oriented architectures is that service interactions occur when two or more services exchange messages with each other following a specific protocol to achieve a particular effect, which is called real world effect62. A message exchange

---

[62] The OASIS Reference Model OASIS-SOA-RM (see Footnote 59) speaks of *real world effect* as the particular purpose associated with interacting with a service.

implies that the sender and receiver understand the syntax and semantics of a message and support mechanisms to send and receive messages. While the message and its payload are specified by Information Models, exception conditions and error handling must be specified as part of the service description. According to OASIS-SOA-RA,

- The message conforms to a referenceable message exchange pattern (MEP).
- The message payload conforms to the structure and semantics of the indicated information model.
- The messages are used to invoke actions against the service, where the actions are specified in the action model, and any required sequencing of actions is specified in the process model.

Messages may be distinguished by type: actions that cause a real world effect and events that report a real world effect. When performing an action against a service the resulting real world effect is typically reported by an event. Actions may trigger operations. Operations are sequences of actions a service must perform in order validly to participate in a joint action, i.e. an action involving the effort of multiple services to achieve a real world effect.

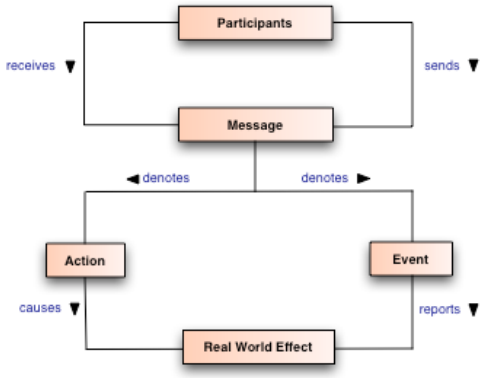


Figure 23: OAISIS-SOA-RA Message Model

Based on these assumptions, two (well-known) fundamental Message Exchange Patterns (MEPs) characterise basic temporal behaviours of service interaction

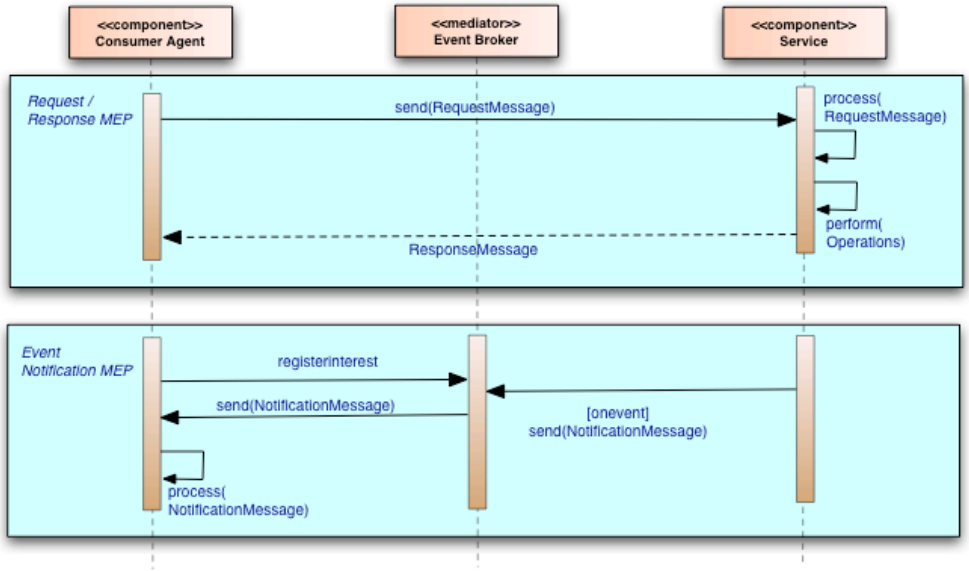


Figure 24: Fundamental SOA Message Exchange Patterns

A service consumer may be client software or another Service Instance (service)63. A Message Exchange Pattern is a template, devoid of application semantics, that describes a generic pattern for the exchange of messages between Service Instances. It describes the relationships (e.g., temporal, causal, sequential, etc.) of multiple messages exchanged in conformance with the pattern, as well as the normal and abnormal termination of any message exchange conforming to the pattern. Different patterns can be observed according to where the focus of the communication characteristics is set.

Pointing at the number and origin of messages exchanges between two participants (consumer and provider) leads to the following eight distinctions, whereby "in" indicates the direction from service consumer to the service provider and "out" the opposite:

- In-Only and Out-Only: One-Way communication where the consumer sends a message to the provider (In-Only, request) without expecting more than a status response or the provider sends a message to the consumer (Out-Only, notification). The W3C WSDL 2.0 specification64 also adds the optional characteristic of robustness to the communication, specifying if fault messages are propagated in the opposite direction to the original message (Robust In-Only, Robust Out-Only)
- In-Out and Out-In: Two messages are exchanged, At In-Out one message is sent from the consumer to the provider (request) and one is sent back to the consumer (response). At Out-In the service provider is the initiator of the communication, but this pattern is less common. In addition, the second message may be optional (In-Optional-Out and Out-Optional-In).

Focusing on the relationship between communication time and process execution between the participants the communication mode can be:

- Synchronous: the consumer interrupts its activities and principally waits for the response (or a timeout) to continue execution (applicable for short time responses, In-Out MEP) or
- Asynchronous: the consumer sends a request and continues. A notification by the service provider informs the consumer that and how a result is available (applicable for operations with long response times, In-Only MEP followed by an Out-Only or Out-Optional-In MEP).

Analysing the number of participants in the communication leads to the following categories of collaborative interaction patterns65:

- Single-transmission bilateral patterns (peer-to-peer communication, refers to the MEP)
- Single-transmission multilateral patterns (One-to-many send, One from many receive and One-to-many send/receive)
- Multi transmission interaction patterns (Multi-responses, contingent requests, atomic multicast notification) An example is a streaming pattern, where the consumer sends a request to the provider and subsequently, the consumer receives any number of responses from the provider until no further responses are required. The trigger of no further responses can arise from a temporal condition or message content, and can arise from either the consumer or the provider 's side.
- Routing patterns (request with referral, e.g. a request is sent indicating that any response should be sent to a number of other parties, relayed request, dynamic routing)

LifeWatch Policy: (i) LifeWatch Service Instances shall interact only by exchanging messages (as opposed to changing a shared state)

---

63 In this section the term service will be used instead of service instance for the implementation abstract description of the pattern, although the real interaction is just possible between service instances.

64 W3C (2007): Web Services Description Language (WSDL) Version 2.0. Part 2: Adjuncts: W3C Recommendation 26 June 2007, available at http://www.w3.org/TR/2007/REC-wsdl20-adjuncts-20070626 (August 2009)

65 For more information about these patterns see http://math.ut.ee/~dumas/ServiceInteractionPatterns/patterns.html and http://www.workflowpatterns.com/documentation/documents/ServiceInteractionPatterns.pdf

(ii) UML Sequence Charts shall be used for specifying formally the temporal or causal properties of Message Exchange Patterns. If other formalisms such as, e.g., Petri Nets or π-calculus are to be used, a respective LifeWatch Policy should be specified.

(iii) LifeWatch Service Networks shall adopt the OASIS Message Model as outlined in this section.

(iv) Message Exchange Patterns to be used by a LifeWatch Service Networks shall be specified.

(v) There shall be an agreed-upon list of Message Exchange Patterns to be supported by all LifeWatch Service Networks. The simple Request/Response pattern shall be in the list. Inclusion of the Notification pattern is an option.

(vi) Synchronous stateless Request/Response shall be the preferred message exchange pattern (reflecting the requirements of Simple Service Architectures)

7.3.4 Protocols

Protocols may have a dual functionality being used either as a pure transport protocol (like HTTP for SOAP) or combine transport with message exchange (as http for REST). There are benefits in both. In the first case, message exchange can be organised generically independent of the particular protocol layer but typically with the disadvantage of additional overhead. In the second case, roughly the reverse holds. No particular recommendation shall be given except

LifeWatch Policy: *(i) Message exchange patterns shall be specified independently of the protocol layer used.*

(ii) In a platform specification, the mapping of message exchange patterns to the particular protocols used shall be explicitly stated.

7.3.5 Service interaction with regard to several platforms

A service network may be mapped onto several service platforms. For instance, one platform may be based on SOAP and another one on REST (see "Status Report on Infrastructures for Biodiversity Research" (Deliverable 5.1.4), Appendix B). Then there are two options for service instances to interact if they reside in different platforms.

- Service Instances may provide several Platform Logics (as discussed above) enabling cross-platform interaction of services.
- A gateway between service platforms is provided that maps between the platforms in a transparent way (e.g. by translating between the formats of messages).

LifeWatch Policy: *(i) LifeWatch shall restrict the use of several platforms.*

(ii) LifeWatch shall prefer the first option (given a small number of platforms).

(iii) The second option shall be used to bridge between platforms that are distinguished by versioning only.

7.3.6 Service coordination

A Business Process describes the tasks, participants' roles, and information needed to fulfil some objective. It is comprised of a set of coherent activities that performed in a coordinated way, taking in account temporal and causal dependencies, to result in a certain outcome. Participants are stakeholders with the capability to act within the framework of a service network such as service providers, service consumers,

service mediators, or agents (behaving on behalf of a person or organisation). In a SOA-world, a business process is achieved by the coordination of services. According to [Barros & al]66, service coordination (or service composition) can be considered from different viewpoints:

- Behavioural Interface – captures the dependencies between interactions as provided or expected from the perspective of a particular service.
- Choreography – refers to the coordination of multiple services by message exchange from a global perspective.
- Orchestration – refers to the typically hierarchically organised coordination of interaction of several services and of internal actions controlled by a single service. Orchestrations may be presented in terms of formally defined workflows that are executed by a workflow enactor.

The viewpoints relate to each other in that an orchestration or choreography defines an expected behaviour interface of the services involved and, vice versa, the interfaces provided specify boundary conditions for orchestrations and choreographies.

LifeWatch Policy: *(i) UML Sequence Charts shall be used for specifying formally the temporal or causal properties of Behavioural Interface and Choreographies. Alternatively, formalisms such as, e.g., Petri Nets or p--calculus may be.67*

(ii) Orchestrations should be specified as workflows the enactment of which is achieved using a workflow engine to control the execution.

7.3.7 Policy frameworks

Policies are about the conditions of use of a service. There are several aspects related to policies: the kind of policy assertions, the ownership of a policy, and the enforcement of a policy. Policies may relate to a single service ("response time is less then 5 seconds"), to a group of services, or to all service instances in a network ("all messages are encrypted"). Policies typically address security, privacy, visibility, manageability, and quality of services. Policy frameworks including the mechanisms for specification and enforcement of policies should be considered as part of a platform specification.

Subsequently, a number of policies are listed that are derived from the general engineering guidelines as well as some policies concerning particular services. The policies statements advance from the general to the more specific.

LifeWatch Policy: *(i) The mapping of the policy frameworks to a particular platform shall be explicitly stated in each platform specification.*

(ii) All participants in a service network shall define the necessary policies as required by the policy frameworks.

Note: The subject of "policy" is self-referential in that LifeWatch policies should be owned and enforced by the participants in LifeWatch.

---

66 A. Barros, M. Dumas, P. Oaks, Standards for Web Choreography and Orchestration: Status and Perspectives, in: C. Bussler et al. (Eds.) *BPM 2005 Workshops*, LNCS 3812, pp. 61 – 74, Springer, 2006

67 One should note that, for choreographies in particular, descriptions may have to deal with the intricacies of specifying (and understanding) concurrent processes.

7.3.8 Levels of conformance concerning technical capabilities

The LifeWatch ICT infrastructure provides technical capabilities as a basis for its biodiversity capabilities (compare Figure 3). Essentially, a LifeWatch service network provides these technical capabilities. While services are the means for accessing all the technical elements of the LifeWatch ICT infrastructure, being by itself "atomic" technical capabilities, only the organized interaction or coordination of services in terms of orchestrations and choreographies finally provide the complex technical capabilities needed in LifeWatch.

Hence, apart from the number of services provided, a service network can be distinguished by capabilities that are supported. Such capabilities may be provided by individual services but may as well by the interaction of several services. Section 7.6 presents several such capabilities in terms of orchestrations or choreographies. The list is not exhaustive. The maturity of a (the) LifeWatch Service Network will be judged in terms of the (core) services provided that are conformant to the LifeWatch Reference Model and in terms of conformance to the orchestrations and choreographies being defined by the LifeWatch Reference Model. Naturally, there may be several levels of conformance and these levels need being defined.

LifeWatch Policy: *(i) LifeWatch shall define which service orchestrations and choreographies are considered inherent part of the LifeWatch ICT infrastructure. The list of these may be extended in the construction phase.*

(ii) In the construction plan, LifeWatch shall define levels of conformance to determine construction steps to be taken.

7.3.9 Federated service networks

A federation is a collection of organizations that agree to interoperate under a certain rule set. It consists of distinct and autonomous components spread across administrative domains being defined by the participating organisations. The components operate independently, but have given up some autonomy in order to participate in the federation. Participation in a federated system allows access to potentially unique or large sets of resources such as data, storage space, computing power, or services.

It is anticipated that the LifeWatch ICT infrastructure will be a federation of research infrastructures being grounded nationally and/or thematically. There are two possible scenarios:

- Service instances of a service network belong to different administrative domains (Federated Service Network)
- Several service networks coexist that belong to different administrative domains (Federation of Service Networks)

For instance, it may be reasonable to distinguish between a service network for data providers and one for users both of which form a federation of networks. One should note that each network within a federation itself might be a federated service network with administrative domains organised, e.g., by subject and nationality.68

The difference between a federated service network and a federation of service network is minor in case that all the service networks conform to the LifeWatch Reference Model. Then only service interaction

[68] It is a question whether LifeWatch will support more than one Service Network. Existence of only one uniform LifeWatch Service Network might be the preferred option. However, given the multi-national structure of the LifeWatch project, the possibility of several networks emerging cannot be dismissed. Hence, a LifeWatch strategy for such a case should be defined, as provide in this section.

has to be guaranteed as discussed in Section 7.3.5. Note that there may be different levels of conformance (Section 7.3.8). Hence, the level of conformance of a Federation of Service Networks may be defined as the minimal level all service networks in a federation conform to, obfuscating the distinction between a Federated Service Network and a Federation of Service networks in that, with regard to the respective conformance level, a Federation of Service Network behaves like a Federated Service Network.

In any case, an agreement has to be reached about the extent of the federation. The extent may vary depending on the level of conformance. A minimal requirement seems to be that a federation should support federated identities, i.e. enable portability of identity information across the boundaries of administrative (security) domains. Similarly, access to data of similar content in different administrative domains should be granted (Federated Feature Access) as well as to catalogues or registries (Federated Catalogue Service).

LifeWatch Policy: *(i) LifeWatch aims for a Federated Service Network with regard to core services and the related orchestrations and choreographies but may otherwise be with a Federation of Service Networks. In particular, it is expected that Thematic Service Networks will be part of a Federation while internally behaving like Federated Service Network (with different organisational and/or national administrative domains).*

(ii) The level of conformance to be attained and the respective timescales shall be defined as part of the construction plan.

(iii) Basic core services and selected (to be defined) support core services shall be provided as federated services.

7.4 Service description

In OASIS-SOA-RA, a Service Specification (or Service Description69) defines the information needed to use, deploy, manage, and otherwise control a service. It is modelled as a subclass of general Description class that is a subclass of a Resources class. Resources are owned by Stakeholders. A description70 describes one or many resources while referencing the respective identifier(s) for an identity. As a Resource, Description has an identifier with a unique value for each description instance that allows the description itself to be referenced. The general Description class is composed of elements that are expected to be common to all its subclasses. These include associated annotations, categorisations, and provenance information.

For convenience, the OASIS class diagram for service description introduced in Section 6.4 is repeated below

[69] OASIS-SOA-RA uses the term “Service Description”. Here “Service Specification” is used to be consistent with the ORCHESTRA terminology.

[70] For classes, capital letters are used, for instances small letters, e.g. ‘description’ refers to an instance of the class ‘Description’.

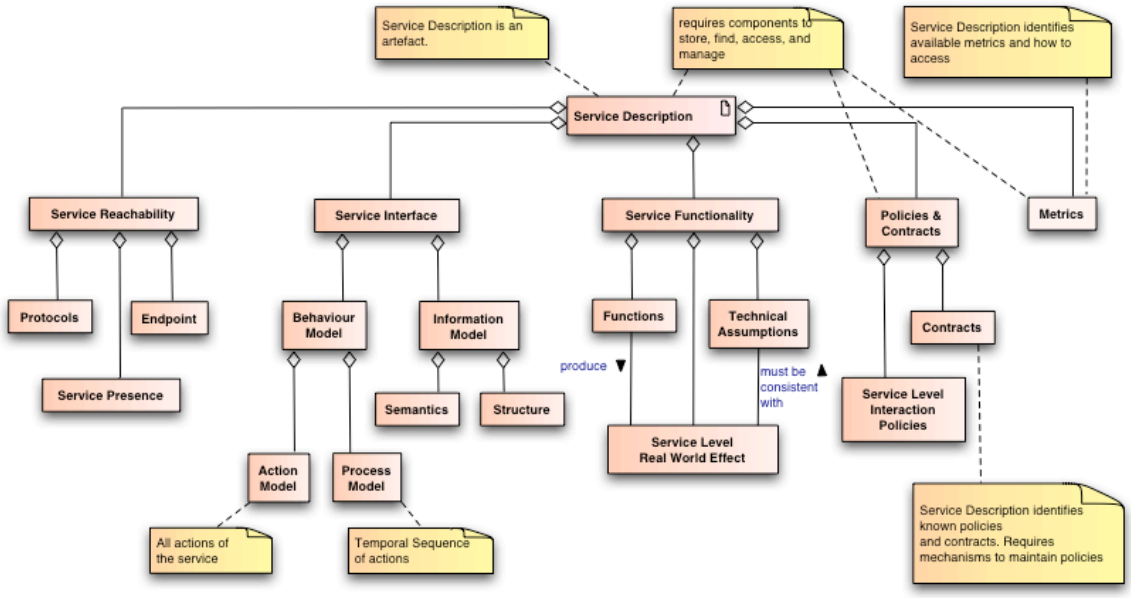


Figure 25: OASIS-SOA-RA Service Description

The elements of a Service Specification relate to platform-neutral or platform-specific level or to both.

LifeWatch Policy: *(i) Each element of a Service Description shall take place at the most abstract level, i.e. platform-neutral. The description of an element may be split if it has platform-neutral and platform-specific aspects.*

(ii) A Service Description should in general consist of two parts, an abstract (platform-neutral) Service Specification and an (platform-specific) Implementation Specification (preferably) in terms of two separate documents.

(iii) The Service Functionality shall be part of the abstract specification.

(iv) The implementation specification of a service should provide the service description in terms of service interfaces descriptions, service endpoints (for reachability), service policies, and service contracts, e.g. as a WSDL document. Elements of service descriptions can be provided through references (links) to external sources to facilitate the reuse of standards for functionalities and policies, among others.

(v) Abstract and Implementation Specifications are required for all basic core services and recommended for support core services

7.4.1 Service functionality

Service Functionality describes what can be expected when interacting with a service in that it defines the interfaces, operations, and parameter types of the service syntactically and semantically. The semantics may be provided I terms of a textual description or in terms of a more formal knowledge capture. The operations produce the desired Real World Effects. The Service Functionality may be constrained by Technical Assumptions related to the underlying capability accessed by the service.

In LifeWatch, Service Functionality is specified in terms of (references to) interfaces and operations, associated types, semantics, policies, choreographies.

LifeWatch Policy: *(i) Interfaces and associated types shall be specified by UML class diagrams.*

(ii) UML class diagrams should be provided as XMI files.

(iii) Some template for textual specification shall be used. It is recommended to use the ORCHESTRA template for abstract service specification.

7.4.2 Service interface

Note The Service Interface should not be confused with Interfaces. The latter are used to structure Service Functionality while the Service Interface defines the relation between the service and the communication network.

The Service Interface is the means for interacting with a service. It includes the specific protocols, commands, and information exchange in terms of messages by which actions are initiated that result in the real world effects as specified through the Service Functionality portion of the service description. The Information Model determines the structure and semantics of messages while the Action Model of a service is the characterization of the actions that may be invoked against the service and the message exchange pattern used to perform an action. The Process Model captures the behavioural aspects of the interactions in which a service can engage to achieve a (business) goal. This includes, for instance, the temporal relationships and temporal properties of actions and events. The Process Model focuses on process behaviour as provided by one particular service in terms of behavioural interfaces. A service may engage in orchestrations and choreographies that may be part of the process model or may be part of the platform specification and referenced in the process model. Orchestrations may also be specified outside the service specification, for instance as workflow description. The Service Interface is implemented by the Platform Logics of an Implementation Object in that it must satisfy the requirements expressed by the Service Interface specification.

LifeWatch Policy: *(i) All actions consist of an operation call applied to arguments of suitable number and type. The Action Model is implicitly specified by the interfaces as defined by Service Functionality.*

(ii) For each action, the message exchange pattern is specified that is used to perform the action. This is required in the implementation specification but may already indicated by the abstract specification.

(iii) While the structure of and protocols for messages are defined on platform level, the data exchanged conforms to the LifeWatch Information Models

(iv) Actions shall describe which effects can follow in terms of shared-states, which are always visible to the service consumer (Real world Effect). The execution context of an interaction must be identifiable.

(v) The Process Model abstractly specifies the process behaviours of a service in terms of Message Sequence Charts (or other formalisms eventually to be agreed upon). These will be part of the abstract service specification.

(vi) Other constraints on process behaviours such as time constraints or exclusive resource allocation are specified within the implementation specification.

(vii) The Process Model references the message exchange patterns, orchestrations, and choreographies the service is engaged in.

(viii) All services instances should provide uniform access to all information and meta-information models associated to it including the information and behaviour models, and provenance through the Capabilities interface.

(ix) Service consumer and service provider must have a consistent interpretation of the semantics of a service information model. The service interface should unambiguously reference to the definitions of the elements used in the information models, for instance through a reference to a domain ontology.

7.4.3 Service reachability

Service Reachability as modelled by Figure 26 enables service participate to locate and interact with one another. An Endpoint is the logical location of a service to which a message can be send according to some protocol in order to invoke an action. Reachability depends on presence of a service and its actions, which may be a static as well as dynamic property.

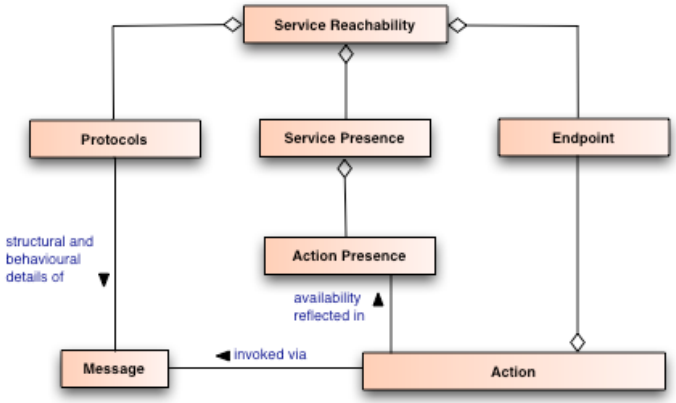


Figure 26: OASIS-SOA-RA Service Reachability Model

LifeWatch Policy: *(i) A service description shall provide sufficient information to enable the interaction of service consumers and service providers.*

(ii) This shall include information about the endpoint(s) supported by a service and about the protocols it supports.

(iii) The service description should provide information about how to check for the presence of a service and its actions.

7.4.4 Policies & contracts

Polices and contracts are part of the service description. Service contracts should be comprised of functional components such as type, functionality, operations, invocation methods, location, pre- and post conditions, and participation in orchestration or choreographies as well as non-functional components such as security constraints, Quality of Service, semantics, transactional properties71, and service level agreements. In that, contracts and policies constitute meta-information about a service.

The fundamental part of a service contract consists of the service specification documents that express its technical interface. These form the technical service contract that essentially establishes an API into the functionality offered by the service. There are mainly two approaches for service development: Turning existing code (automatically) into a technical service contract is called a code-first approach. By contract-first approach, the contract is developed first and the code is often generated automatically.

LifeWatch Policy: *(i) Policies and contracts or references to these are part of the service specification.*

(ii) Policies and contracts must be available in a machine processible format.

(iii) Services should be developed by contract-first approach.

---

71 Whether it capable of acting as part of a larger transaction.

7.4.5 Reusing ORCHESTRA service descriptions

ORCHESTRA provides a number of specifications72 of core services that, in parts, comply with the policies outlined above.

LifeWatch Policy: LifeWatch will reuse the ORCHESTRA service specification and adapt as necessary.

Note: User management, authentication, and authorisation services cannot be reused as are since LifeWatch will, in contrast to ORCHESTRA, use single-sign-on for authentication (see Section 7.7.6).

7.5 Ownership perspective

Before a service network is set up for running, a series of requirements and responsibilities related to the governance of the network, as well as security and management challenges must be considered. Governance is concerned with decision-making, security with access control, and management with the execution of a Service Network.

This section addresses these three issues and relates them with operating policies for LifeWatch Service Networks. The components of a service network, e.g. the interacting services, may be under the control of different ownership domains. The operating policies ensure a standardisation across internal and external organizational boundaries.

7.5.1 Governance

According to the OASIS-SOA-RA, governance of a SOA based system must address the following questions in order to achieve specific goals, add value, and reduce risk:

- What decisions must be made to ensure effective management and use?
- Who should make these decisions?
- How will these decisions be made and monitored?

One of the primary topics for governance is how to obtain acceptable terms and conditions in a service contract. Participants of a service interaction are not just the service consumer and provider, but also the owner of the underlying capabilities that the service access. A governance model must establish the rules and policies under which duties and responsibilities are defined and the expectations of the participants are grounded. Participants can express their expectations in terms of transparency, trust, and reliability.

Figure 27 describes governance elements and their roles in order to construct a governance model. The overall governance strategy should be consistent with the goals of the participants. Governance requires an appropriate structure and identification of an authority, which makes the decisions. The leadership represents this authority and is responsible for initiate and control governance, through generation of consistent policies, expressing the governance rules and regulations. The participants should recognize the authority of the Leadership, who typically has some control over the Participants. The Leadership prescribes or delegates a working group to define the Governance Framework that forms the structure for the Governance Processes. The Governance Framework should enable flexibility indicating a combination of strict control over a set of foundational aspects, as the definition of well-behaved services. To ensure well-behaved services sufficient metrics are needed to know how the service affects the infrastructure and whether it complies with the established policies. The Governance Processes define how governance is to be carried out by providing an unambiguous set of procedures, which are likely reviewed and agreed by the Participants.

---

72 http://www.eu-orchestra.org/publications.shtml#OAspecs

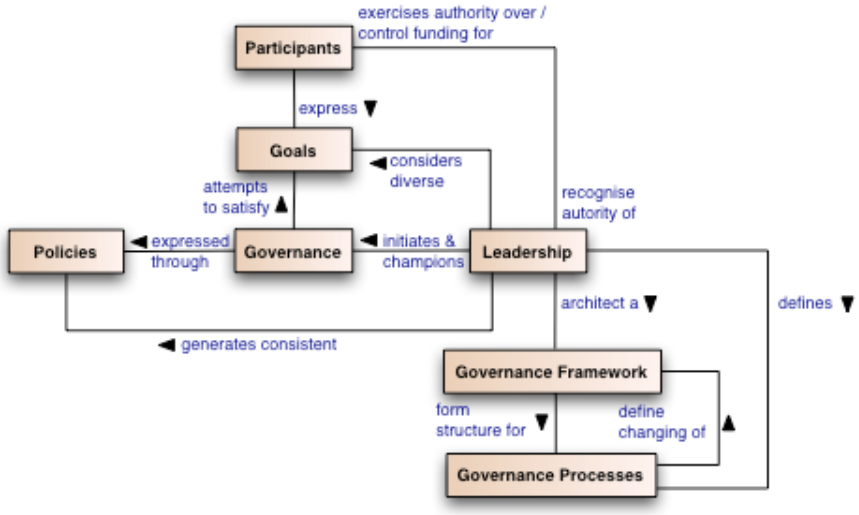


Figure 27: Setting up a Governance Model according to OASIS-SOA -RA

LifeWatch Policy: *(i) The LifeWatch Governance is expressed through policies and policy descriptions, accessible for all LifeWatch Participants, which are the LifeWatch Stakeholders. Example of Participants are the LifeWatch Service Centre, the LifeWatch Central Staff, service providers, data providers, tool providers, and related organizations (Networks of Excellences, Projects, Standardization Bodies).*

(ii) The LifeWatch Infrastructure should support that Participants understand the intent and the structures of the governance model in order to make governance operational. The LifeWatch Portal should provide access to the governance policies as text, e.g. as part of the LifeWatch Reference Model, the LifeWatch Construction Plan, as well as references to the Standard Specifications underlying the governance framework. It is recommended to provide a discovery mechanism, where Participants can search for policies that best meets particular criteria, and a mechanism to inform participants of significant governance events, e.g. though offering a syndication mechanism.

(iii) Service providers should have access to implementations of governance processes for including them into the services.

(iv) The rules and regulations that make governance policies operational should be unambiguously identifiable and version controlled.

(v) Governance relies on metrics to define and measure compliance. The LifeWatch infrastructure should provide automatic test mechanisms to facilitate the proof for compliance and interoperability. A compliance-testing program[73] should be developed during the construction phase. The LifeWatch compliance-testing program should develop tools to test general conformance policies like interoperability, usability, accessibility, and multilingualism, and support the conformance test to service specifications and standards.

All Abstract Service Specifications should be accomplished of a normative Abstract Test Suite, describing how to achieve syntactically, semantically and other conformances to the specification. Implementation Specifications of Core Services should be

[73] As example can be see the OGC Compliance Testing Program (http://www.opengeospatial.org/compliance) and the Online testing site developed by the Compliance & Interoperability Testing & Evaluation (CITE) Initiative (http://cite.opengeospatial.org/)

accomplished of automatic test suites and their descriptions to be included into an automatic testing system.

7.5.2 Security

Security is one aspect of confidence in the internality, reliability, and confidentiality of a system. It focuses on avoiding inappropriate access to resources and on accidentally or intentionally misuse of the infrastructure. According to the OASIS-SOA-RA, security can be defined in terms of the "social structures that define the legitimate permissions, obligations, and roles of people in relation to the system, and mechanisms that must be put into place to realize a secure system".

ISO/IEC 27002 characterizes the following key security concepts:

- Confidentiality: Protection of privacy of participants in their interactions: messages should not be readable to third parties, but the degree of visibility if messages are exchanged and of the participant's identity to third parties can be defined
- Integrity: Protection of altering exchanged information
- Availability: Concerns the reliability of a system, in other words, if the system offers the service for which it is designed, and the security concept needed to respond to active threats to the system
- Authentication: Concerns the means of identifying the participants in an interaction
- Authorisation: Concerns the means of legitimacy of the interaction, the exchanged actions must be explicitly or implicitly approved
- Non-repudiation: Concerns the accountability of participants: participants should not, at a later time, successfully deny having participated in the interactions.

Although a system will never be able to guaranty these security goals to 100%, a well-designed security model can ensure acceptance levels of security risks.

A security model can be described at three layers:

- At the Network Layer, security is expressed in terms of availability of the services and protection of Denial of Service attacks. On the network layer, general policies or Service Level Agreements (SLAs) about the quality of services can be defined. The Network Layer is beneath the scope of the Reference Model. All Service Networks are assumed to fulfil some general policies.
- At the Transport Layer, security is concerned with the establishment of secure communication channels between sender and receiver. It may include protocols like the HTTP over Transport Layer Security (TLS)74 or HTTP over Secure Sockets Layer (SSL). The security model for the Transport Layer is part of the platform level definition.
- At the Application Layer, security is applied to the messages exchanged between participants. Web Service Standards addressing security concepts at this layer are WS-Security, XML Digital Signature, XML Encryption, and SAML. The security model of the application layer may be part of the description of the Service Network, but can be specialized for particular applications running inside the Service Network.

The security model comprises three models:

[74] TLS is a cryptographic protocol standardized by the IETF, last updated in RFC 5246 (http://tools.ietf.org/html/rfc5246) , that was based on the earlier SSL specifications (http://www.freesoft.org/CIE/Topics/ssl-draft/3-SPEC.HTM)

- The Trust Model ensures to identify and validate the participants and their interactions in the system. It is depicted in Figure 28 and contains the following concepts, which are related to the ORCHESTRA UAA Concepts75 as follows:
    - Participant: Represents a user or a software component involve in an interaction. The ORCHESTRA Reference Model represents participants as Subjects.
    - Trust: Relationship perceived by a stakeholder between a participant and a set of actions and events, which concerns the legitimacy of the actions and events.
    - Credentials: Roles and/or set of attributes a stakeholder uses to determine authorisation to actions of a participant.
    - Identity: Identifies a participant. It is used together with credentials to provide suitable authorisation before continuing the interaction. The ORCHESTRA Reference Model represents identities as Principals. Each subject may have more than one principal.
    - Trust Domain: Abstract space of actions which all share a common trust requirement, i.e. all participants must be in the same trust relationship. At an application level, a trust domain may refer to a social structure, e.g. a user group. The ORCHESTRA Reference Model refers to the concept of Group, as a specialization of subject, which can have one or more principals (group principals) identifying the group. Different principals can be assigned to a group as member principals, which are the particular identities of all subjects belonging to the group.

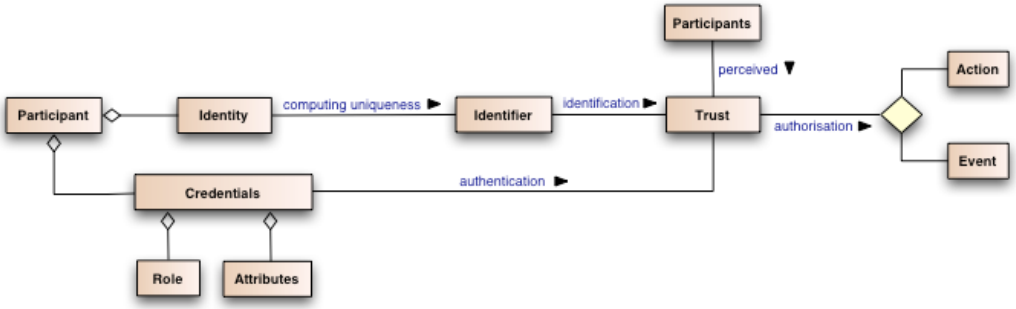


Figure 28: OASIS-SOA-RA Trust Model

- The Threat Model can be used to define a list of common threats related to the core security concepts, e.g. for message alteration, message interceptions, denial of service attack, spoofing attacks, etc.
- The Security Response Model defines levels of risks based on threat assessments and the costs of mitigating those treats. Possible mechanisms are:
    - Usage of information encryption to assure confidentiality
    - Usage of digital signatures to ensure integrity
    - Inclusion of a message ID, timestamp, and destination to a message to ensure that each messages ever sent is unique, for protection against replay attacks
    - Maintenance complete logs of interactions, usable for auditing purposes to avoid false repudiation

LifeWatch Policy: *(i) Security policies require mechanisms to support security description administration, storage, and distribution. These mechanisms shall be defined within LifeWatch.*

(ii) Security policies should be able to express trust relationships and trust domains, providing the ability to update trust relationships without changes in the hard- and software.

---

75 See Section 7.5 User Management, Authentication, and Authorisation on the ORCHESTRA Reference Model, which is summarized on Appendix B.5.4.9.

(iii) Standard Protocols should be used to provide confidentiality, integrity, authentication, authorisation, non-repudiation, and availability.

(iv) Service Specifications and Service Descriptions should be able to reference one or more security policy artefacts.

(v) A Service Network should provide mechanisms for:
- *protection of the confidentiality and integrity of messages exchange*
- *policy based identification, authorisation, and authentication*
- *ensuring service availability to the consumers*
- *ensuring security for a scalable network and between different platforms.*

(vi) A Service Network should include instances of the following security services:
- *Authentication Service*
- *Authorisation Service*
- *User Management Service*
- *Service Monitoring Service for Security including monitoring of intrusion detection and prevention, auditing and logging of interactions, security violations, service availability, and support for quality of services.*

(v) LifeWatch Services should use an Encryption Interface to abstract encryption techniques, allowing the use of different techniques.

(vi) There shall be an agreed-upon list of Security Policies, e.g. for the Network- and Transport-Layer to be supported by all LifeWatch Service Networks. These policies may also include generally valid aspects of the three security models (trust, treat model, and security response).

(vii) A threat model shall be defined in form of a list of exceptions thrown by the Service Monitoring Service.

7.5.3 Management

Management is concerned with controlling the use, configuration, and availability of resources in accordance with the policies of the stakeholders involved. This involve three aspects:

1. Management of all resources involved in the Infrastructure/ Service Network.
2. Publication and enforcement of the policies and contracts agreed to by the stakeholders.
3. Management of the relationships of the participants.

Management may reflexively be considered as part of the service-oriented architecture in that a Management Service may support it. A management service is meant to manage all the resources of a service-oriented architecture. For being managed, a resource needs to expose its manageability capabilities to the management service. According to OASIS_SOA_RA, manageability capabilities include

- Lifecycle Manageability – is typically concerned with creation and deletion of a resource and handles the relation to other resources that must coexist.
- Configuration Manageability – for configuring a resource.
- Event Monitoring Manageability – supports, for instance, reporting of tracing events and faults.
- Accounting Manageability – for measuring and accounting for the use of resources by properly identified participants.
- Quality of Service Manageability – for managing quality of service requirements for a resource, typically a service.
- Business Performance Manageability – for monitoring and managing business agreements like SLAs (service level agreements).
- Policy Manageability – for handling policies, e.g. promulgation and enforcement.
- System Manageability – concerning the administration and maintenance of computing resources

- Network Manageability - concerning the administration and maintenance of the network resources

There may be different management services addressing different aspects. As other services, management services are subject to policies and contracts.

LifeWatch Policy: *LifeWatch shall provide management services to support the elements of a basic management service infrastructure as defined following OASIS-SOA-RA:*
- Integration with existing security services
- Monitoring
- Heartbeat and Ping
- Alerting / Notification
- Pause/Restore/Restart Service Access
- Logging, Auditing, Non-Repudiation
- Runtime Version Management
- Complement other infrastructure services (discovery, messaging, mediation)
- Message Routing and Redirection
- Failover
- Load-balancing
- QoS, Management of Service Level Objects and Agreements
- Availability
- Response Time
- Throughput
- Fault and Exception Management

7.6 Meta models

ORCHESTRA provides a metamodel, parts of which are presented in Appendix B. This metamodel is not very specific in that features as, e.g., service description, are of type CharacterString. It may be refined to capture more precisely the types and in that the requirements for the specification of models. Possibilities for adaptation and refinement will be discussed in Appendix B.

LifeWatch Policy: *(i) LifeWatch shall adopt the ORCHESTRA Meta Models.*

(ii) LifeWatch will adapt the ORCHESTRA Meta Models to stipulations concerning the description of the entities of the LifeWatch architecture, for instance by reflecting the OASIS SOA Reference Architecture.

7.7 LifeWatch technical capabilities

The technical capabilities of LifeWatch consist of the services provided and the coordinated use of these. The engineering of LifeWatch services is largely determined by the Service Viewpoint (Section 6) and by Section 7.4 on service description, the notable exception being Source System Services. When analysing engineering aspects of how services can be composed in order to achieve value added functionalities or capabilities, certain choreographies or patterns are relevant. Architectural patterns express a general reusable structural organization or schema for a set of services and their interaction in order to solve a commonly occurring business process or choreography. In the literature, one may find different pattern terms[76], which may be differentiated in the level of abstraction. Architectural patterns are "high-level

[76] According of the description of architectural pattern, based on the ISO 19119:2005 definition of architecture patterns, terms like "architecture pattern" (ISO 19119:2005), "design patterns" (Wikipedia), "SOA-patterns" (Ciurana, Eugene (2009): SOA Patterns), or "service interaction patterns" (ORCHESTRA) can be found in the literature.

strategies that concern large-scale components and the global properties and mechanisms of a system"77. They have wide-sweeping implications that affect the overall skeletal structure and organization of a software system.

The LifeWatch Reference Model is already based on common architectural patterns, SOA. However, there are other architectural patterns that provide technical capabilities as needed by LifeWatch:

- Integration of external sources (not LifeWatch conformant sources)
- Distributed implementation of services
- Provision of resource visibility
- Provision of unique naming for resources
- Semantic interoperability
- Controlled access to resources
- Workflows for orchestration
- Generation of provenance data

Note: This list is not exhaustive nor is presentation conclusive. The list should be validated and eventually extended during the construction phase.

7.7.1 Integration of source systems as services

Source System Integration means the process of transforming an External Source System into a Life-Watch Source System, i.e. the transformation a native (non-LifeWatch) system into a LifeWatch Service Instance (within the LifeWatch service network) that provides access to the data and the functionality of the External Source System.

The LifeWatch Service Instance representing the External Source System must be defined according to the rules of the LifeWatch Service Meta-model described above. This, in particular, implies that it is specified in terms of LifeWatch-conformant interfaces. Due to the heterogeneity of External Source Systems, one cannot expect to define a service type with predefined interfaces that covers all these systems. Hence, ORCHESTRA proposes to use the term Source System Integration Service as a general name for the class of services serving this purpose.

- If an external source system can be treated as a specific instance of an existing LifeWatch service type, the interfaces of this LifeWatch service type shall be implemented.
- New service types shall be specified using existing LifeWatch interface types whenever they are suitable.
- If operations of the external source system do not comply with operations of any given interface, either an available interface type can be extended or a new interface type can be created.
- A mechanism for extracting the meta-information needed by LifeWatch should be implemented, as a publishing mechanism for the external source system.
- The ServiceCapabilities interface must be implemented.

Note: Note that these steps have to be taken by the providers of external source systems as part of the admission procedures for joining the LifeWatch infrastructure.

The main integration challenges are:

- Transformation and integration of existing data models into LifeWatch information models, preferably keeping the original data models, for instance through defined Extraction-Transformation-and Loading (ETL) processes

[77] See Pattern reference site: http://www.cmcrossroads.com/bradapp/docs/patterns-intro.html

- Transformation of meta-data and data into purpose oriented meta-information, according to the LifeWatch models
- Support the invocation of external source services through interfaces of standardized LifeWatch service types
- Meet the AAA-requirements from both the LifeWatch infrastructure and the external source systems.

There are well-known design patterns to be used for source system integration services (e.g. the Adapter and Façade pattern78). The INSPIRE Network Service Architecture Draft79 proposed the use of the façade pattern, as a mediator layer between the INSPIRE Service Bus and existing services, that are not conform to the INSPIRE Implementation Rules. The INSPIRE Service Bus is defined as the backbone of the INSPIRE infrastructure, prescribing the standardized interfaces accessible by consumers of INSPIRE services. . Floczyk et al.80 propose a methodology to make easy building wrappers for existing services to conform to INSPIRE meta-information models. Firstly, a service type target, which defines the templates for meta-information and information models, is selected. Secondly, an appropriated adapter, for meta-information and service descriptions, is chosen if available. After it, the meta-information must eventually be merged to conform to the requirements. Another proposition for INSPIRE is the establishment of an INSPIRE Fusion Centre81, where external data models and web services are transformed and integrated into a data repository optimised for INSPIRE specifications. A final solution has not been decided yet. Note that these steps have to be taken by the providers of External Source Systems as part of the admission procedures for joining the LifeWatch infrastructure.

Example: Inclusion of external source systems

The Feature Access service type (not documented in this paper) can be used for access to biodiversity data using the TAPIR protocol. This can be achieved, for example, by a wrapping mechanism as indicated in Section 4.3.13 plus extraction of the relevant meta-information, as shown in Figure 29 and explained below.

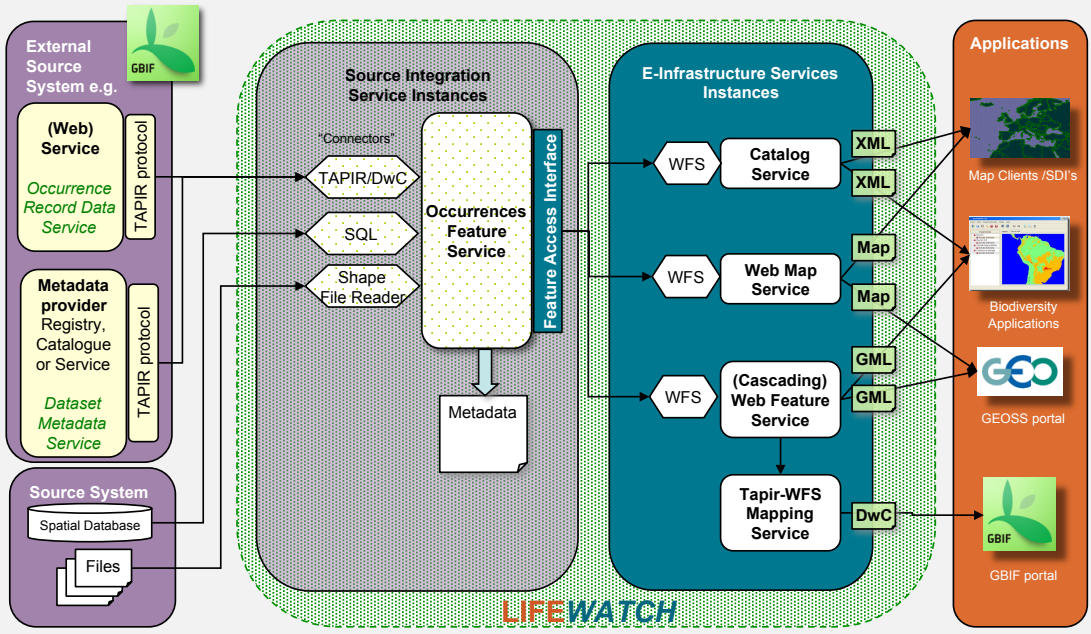


Figure 29: Example inclusion of external source systems

The example shows an instance of an Occurrence Feature Access service at the source integration services (see Figure 7). The Occurrence Feature Access service is an implementation of a Feature Access Service Type and provides therefore the common interface of this service type (Feature Access Interface). Implementation of Feature Access Services provides one or more connectors to specific data sources in specific formats. In the example there are three connectors: a SQL connector for spatial databases, a shape file reader to access ESRI82 data files, and an interface conforming to the TDWG access protocol (TAPIR), with DarwinCore (DwC) as the exchange format. The service knows the data models obtained by the connectors and can extract feature and metadata information from it. LifeWatch services (e.g. catalogue, web map and other feature services) from the Infrastructure (services) layers (Figure 7 again) can then access this data via the Feature Access interface. They, in turn, offer the information to the applications via standard interfaces.

Transformation services, such as the Tapir-WFS Mapping Service shown in the example, allow clients of a particular domain (e.g., GBIF) to use a specific protocol, which can be transformed into standard interfaces.

7.7.2 Distributed implementation of a service

Many LifeWatch services will be offered by different institutions or by different sites. For instance, there may be several catalogue services that, however, should be accessible like one big catalogue without the user being aware of the distributed, may be even federated, nature of the "real" catalogue services. Problems of federation can be resolved by using federated identities (see Section 7.7.6). For distribution, particular schemes need to be defined.

---

[78] See Hohpe, G. and Woolf, B. (2003): Enterprise Integration Patterns. Addison-Wesley, October 2003

[79] Network Services Drafting Team. INSPIRE Network Services Architecture, July 2008.available under http://inspire.jrc.ec.europa.eu/

[80] See. Floczyk, A. J.; López-Pellicer, F. J.; Valiño, J.; Béjar, R.; Muro-Medrano, P.R. (2009): INSPIRE-able services. In Proceedings of AGILE 2009: 12th AGILE International Conference on Geographic Information Science: Advances in GIScience. Hannover, Germany, 2-5 June 2009, available at http://iaaa.cps.unizar.es/curriculum/09-Otras-Publicaciones-Congresos/cong_2009_AGILE_Inspire.pdf (October 2009)

[81] See Elfers, C. (2009) Slides to INSPIRE-Challenges and Solutions for Data and Service Provides, presented at the GSDI 11 World Conference, Rotterdam, June 2009. Available at http://www.gsdi.org/gsdiconf/gsdi11/slides/tues/1.4e.pdf

[82] The ESRI shapefile is a popular geospatial vector data format, describing geometric and thematic attributes. (see http://www.esri.com/)

A candidate scheme is a variant of MapReduce83. The scheme consists of two steps:

1. Map step - A master node broadcasts a request for execution of a particular operation (for instance, a query) to all "worker" nodes. The worker node executes the "mapping" operation on its local data set, and passes the answer back to its master node.
2. Reduce step - The master node collects the answers and combines them so that the result is that obtained if all data would reside in one place.84

The requirement is that instances of mapping operation are independent of each other, hence can perform in parallel. Overall, there should be no distinction between a service operating on an aggregated local version of all the data and a service implemented as described above. Such a distributed implementation of a service may have several master nodes as endpoints in order to avoid bottlenecks due t a single endpoint.

There are many other schemes for distributed implementation of a service and it depends which schemes serves best the needs of a particular service. Since the LifeWatch infrastructure will be distributed (and federated), appropriate implementation schemes have to be chosen.

LifeWatch policy: LifeWatch shall specify which services need a distributed implementation and which distribution scheme is applied.

7.7.3 Provision of resource visibility

A key requirement for service networks is that resources, before they can interact, they have to be visible to each other using appropriate means. A resource consumer has demands or goals and wants to solve them by interacting with a resource or service. The questions are

- How to find suitable services?
- How to know if the service can fulfil the consumer goals?
- How to establish an interaction with the service?

Using the concepts of the OASIS-SOA-Reference Model, service visibility or, more generally, resource visibility can be analysed in terms of awareness, willingness, and reachability, as shown in Figure 30. A resource consumer must be direct or via a mediator with knowledge of the resource descriptions aware of the existence and usage of the service, The consumer is willing to use the service if it assumes, by getting the service descriptions, that the service capabilities satisfies its goals. The reachability of the service by the consumer is accomplished when the consumer is capable of interact with the service. The interactions will result in a real world effect, which will induce to a change of the internal state as is visible to the consumer.

83 http://en.wikipedia.org/wiki/MapReduce

84 This is a variant of the original algorithm in that the data a distributed a priori while the distribution of algorithm and data are part of the map step in the original algorithm.

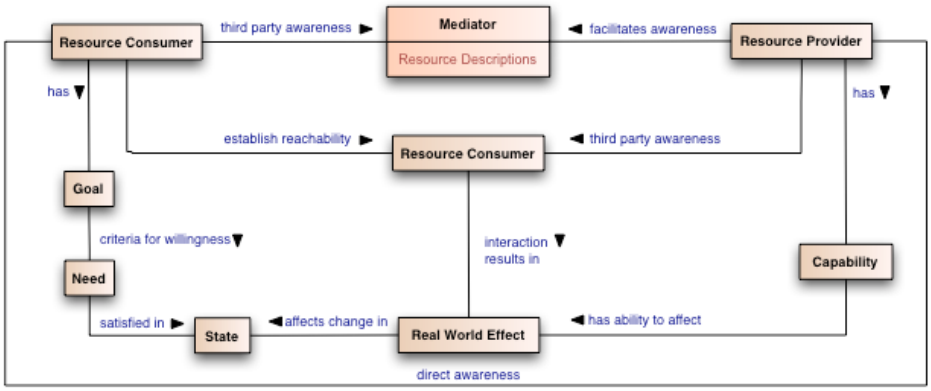


Figure 30: Derived from the OASIS-SOA-RA Visibility Model

Following the SOA concepts the architecture must provide three fundamental operations, often called the publish-find-bind paradigm: publishing a resource to a mediator allows a resource consumer to find the service. Once found, the resource consumer can bind the resource, which means to invoke it or have access to it.

Resource visibility is achieved when a resource consumer can find descriptions of a resource, so that it can access the resource. Therefore, the descriptions must have been made available to a search engine. This process is known as publishing, the search engine is often called Resource Broker or Service Broke and what is made available are meta-information or rather a link to the meta-information about the resource (which should comply to a standard or a format imposed by the resource broker). A resource can be publicised in an active manner if the resource provider or another network component (e.g. the service instance providing the access to the resource or another service instance or organisation) add or modify meta-information to the resource broker. This active management is called “push model” or “transaction operation”. If the resource broker itself is the one responsible for collecting meta-information about resources, then this is called a passive management, or harvesting or “pull-model”. On the other side, resource requestors want to query for resources having specific properties or capabilities in a uniform manner, similar to querying a database. This process, where a potential service consumer searches for resources, navigates through the query results, and asks for information about particular resources to a resource broker is known as discovery. Figure 31 depicts the actors and operations involved in the publishing and discovery process.

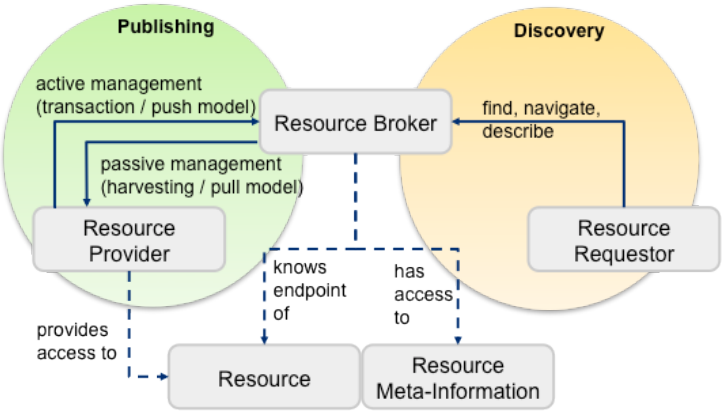


Figure 31: Resource visibility through publishing and discovery

Figure 32 and Figure 33 depicts typical choreographies for resource visibility describing use-cases for publish-find-bind and the distributed search85.

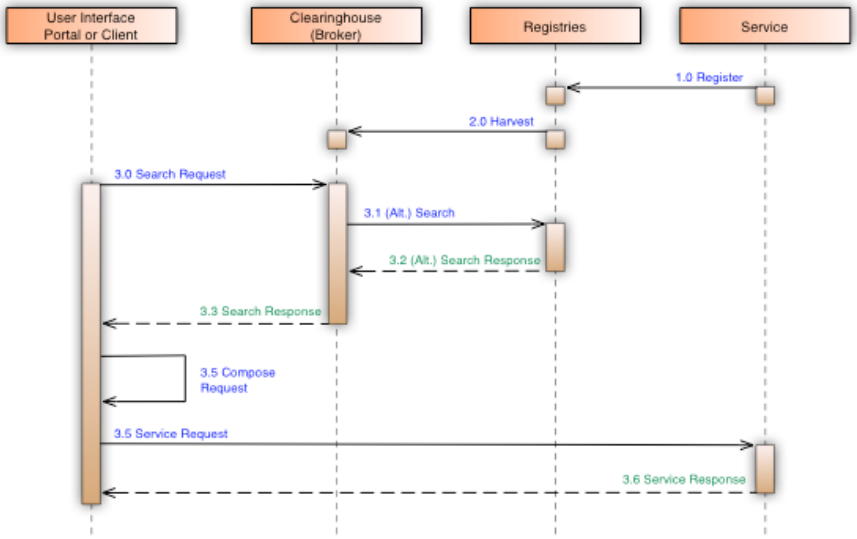


Figure 32: Sequence diagram for publish-find-bind86

The publish-find-bind sequence in Figure 32 reflects three actions: the active registry of a service by a service provider (1.0), the passive management of service descriptions (meta-information) from registries through harvesting, done by the clearinghouse or Broker asynchronously from consumer requests (2.0), and a search request form the consumer (user interface or portal) to the broker (3.0). The first two actions are related to the publish purpose. The last one involves a search, also known as discovery request for services to the broker. The broker can look either in a local cache or, alternatively, forward the request to the registry (3.1) and wait for the response (3.2), before sending a search response to the consumer with the service description (3.3). If the service description fits to the consumer needs, then the find purpose was fulfilled. The consumer can compose a service request (3.4) and directly bind it to the service end-point, sending a message with the service request (3.5) and waiting for a response (3.6).

The main requirements for engineering the discovery architecture to support the publish-find-bind paradigm are (1) how to find actualized service descriptions, considering the big number and distribution of existing services and registries, and (2) how to support discovery with minimum human intervention.

The first requirement concerns distributed search and can be solved by using a broker with access to multiple, cascading catalogues, as well as having two search strategies: one supporting real time search through all the catalogues and the second one using asynchronous harvesting and caching mechanisms to increase performance. This architecture is currently been used in GEOSS and in the INSPIRE EU Geo-Portal). Figure 33 shows a sequence diagram for a distributed search use case provided by the GEOSS implementation pilot. The distributed search is primary performed by the clearinghouse. Harvesting metadata form all catalogues is not practicable for a system of systems, thus it is performed for some entries from some catalogues. In sequence 1.0, for example the clearinghouse harvests just the second catalogue. On a search request (2.0) the internal clearinghouse cache is first searched and the results can be delivered to the requestor (2.2), while the clearinghouse performs further search requests to catalogues (2.3). The Catalogue 1 is a cascading catalogue, which means that the registry entries may refer to other catalogues, so that the search request is forwarded to catalogue 2 (2.5) and composed with the cache

85 The sequence diagrams were adapted from the GEOSS implementation pilot specifications, available under http://www.earthobservations.org/docs/CFP_GEOSS_AR-07-02_11.4.2007.pdf

86 Source: GEOSS Implementation Pilot http://www.earthobservations.org/docs/CFP_GEOSS_AR-07-02_11.4.2007.pdf

search results to a combined search response (2.7). The clearinghouse combines all responses from the catalogues with the own cache search results to a composite response delivered finally to the requestor (2.9).

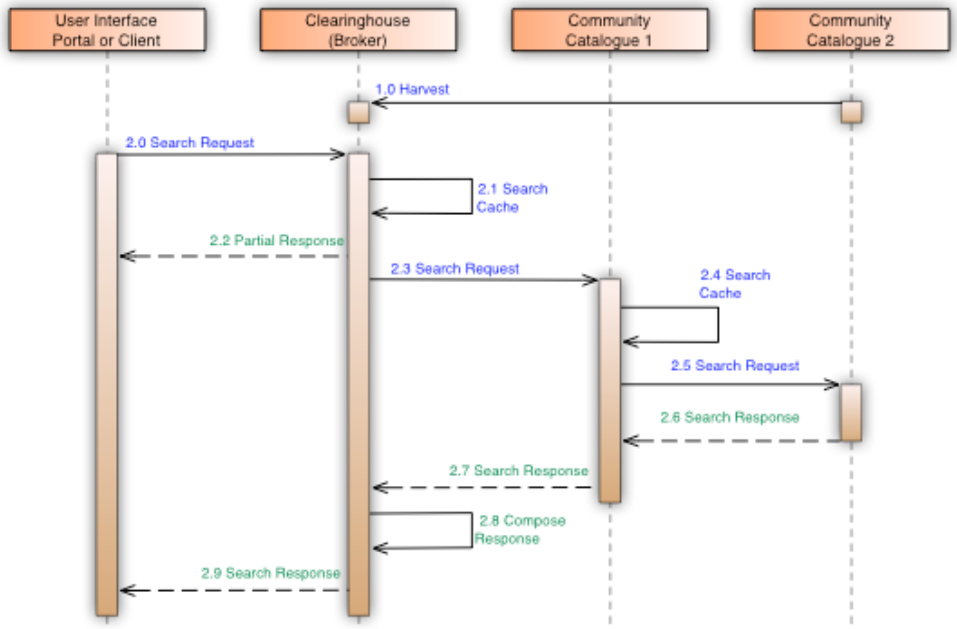


Figure 33: Sequence for distributed search

The distributed search demands interoperability between the catalogues. Experiences building the GEOSS and the INSPIRE catalogue architecture for accessing to existing "standardized" community catalogues, reveal where the difficulties are in and should be considered when building the LifeWatch infrastructure:

- Current specifications (OGC Core, ISO, ebRIM) allow too many degrees of freedom in implementations: interoperability depends largely on implementation compliance and not on specification,
- Specific adapter for certain systems need to be implemented on a case-by-case basis, and
- The semantics of the information models are not well defined, a shortcoming for model mappings.

The second requirement, to minimize user intervention, relates to the shortcoming for semantic descriptions. Semantic heterogeneity arises through synonyms, which are not found through the search engine, and homonyms, which are found but are not appropriate, as well as through missing well-defined vocabularies for information and meta-information models. Many approaches87 follow an ontology-based discovery and retrieval architecture as solution.

---

[87] See Sapkota, B.; Roman, D.; Fensel, D. (2006): Distributed Web Service Discovery Architecture. In Proceedings of ICIW'06 - International Conference on Internet and Web Applications and Services /Guadeloupe, French Caribbean, available at http://dip.semanticweb.org/documents/Brahmananda-Distributed-Web-Service-Discovery-Architecture.pdf (October 2009), Lutz, M.; Sprado, J.; Klien, E.; Schubert, C.; Christ, I. (2009): Overcoming semantic heterogeneity in spatial data infrastructures. Geoscience Knowledge Representation in Cyberinfrastructure. In: Computers & Geosciences, Jg. 35, H. 4, pp. 739–752. Available at http://www.sciencedirect.com/science/article/B6V7D-4RNR6X0-2/2/0e5a73fd1e3a94ad47dfde9d5cf87647 (October 2009), Hilbring, D., Usländer T. (2006) Catalogue Services Enabling Syntactical and Semantic Interoperability in Environmental Risk Management Architectures. In Proceedings of EnviroInfo 2006, available at

LifeWatch Policy: *The LifeWatch Infrastructure should follow a decentralized approach for publishing and discovery. There should be an attempt to follow a fully distributed implementation (each catalogue may act as a broker), but during the construction phase, it should be evaluated if an approach using a centralized clearinghouse is more practicable.*

The LifeWatch discovery architecture should be semantically enhanced.

7.7.4 Provision of unique naming for resources

All resources available at the LifeWatch Infrastructure should be unambiguously identified. Within a service network a specific naming policy should be followed, which must be defined at the platform specification attached to the service network. Nevertheless a service instance may expose different interfaces for different platforms and may change from a service network environment to another one, e.g. due to organisational or administrative issues. Therefore, it is important to keep a track of all identifiers of a resource and the underlying naming policies.

Usually this pattern is solved though a registry providing unique ids in a specific context, when resources register to it. The registered resources should be persistent. Examples or registers are:

- The UDDI Registry that provides a Universal Unique Identifier (a 128-bit number),
- The Digital Object Identifier Registration Agencies that provide persistent identifications (DOI names), or
- Services offering names based on Uniform Resource Names (URN) as they are Object Identifier Registration Authorities (like IANA, ANSI, BSI) which provides hierarchical structured Object Identifiers (OID has a hierarchical structure) or Life Science Identities (LSID) providers like TDWG88.

LifeWatch must ensure that resource naming keeps a dynamic process. For example, service providers can change the organisation and move to another service network with different platform naming policies. ORCHESTRA proposes a strategy for interaction of name services that has been adapted as follows for LifeWatch:

- If a new resource without a LifeWatch compliant GUID is added to a service network A through publication in a registry or catalogue, a name service of the network A should be requested to register the resource and create a valid GUID. The GUID should be attached to the resource meta-information, at least on the registry or catalogue
- If a new resource is added to a service network A with a valid GUID, which was build through a name service of a service network B, then the registry or catalogue sends a request to a name service in network A to create a link to the name service in network B for the resource. By doing this, the name service in network A acts as a cascading name service (see Figure 34).

---

http://www.eu-orchestra.org/docs/20060906-OrchestraPaper-EnviroInfo2006-CatalogueServiceApproach-updated.pdf (October 2009)

[88] For a more explicit description of the standards and the techniques behind them, please refer to the LifeWatch Status Report Document.

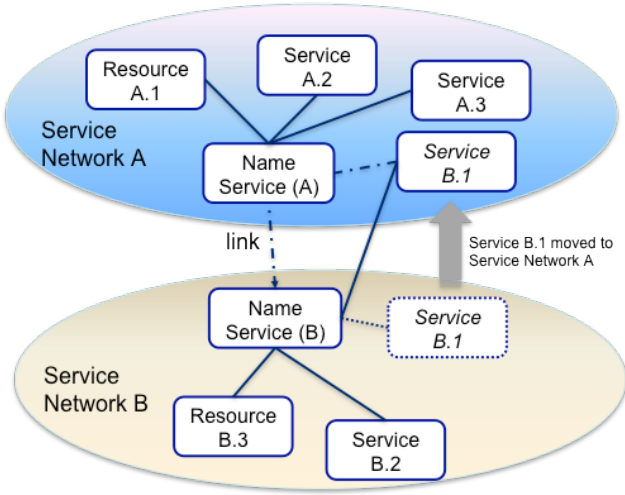


Figure 34: Linkage between name services, adapted from ORCHESTRA-RM

- If a resource is removed from a network A, and the name service does not has any links to other name services for this resource, than the resource can be deregistered from the name service, and the GUID may be deleted from the resource meta-information. Eventually a strategy of keeping the GUID and the entry on the name service for provenance issues should be followed instead.
- A service network may use own name service instances or may reuse existing ones of another service networks
- If a complete service network is removed, name service instances may still be retained for use by other service networks.

A service network providing unique naming for resources need to solve 3 main tasks, which are depicted as choreographies at Figure 35:

(1) The first task concerns the creation of a unique name or identity for a resource during the publication process. The resource provider may actively publish the resource, for example a service (hence the term "service provider" is used in Figure 35): the service provider requests a resource broker (e.g. a catalogue or registry) to publish the service by providing the adequate meta-information (1.0). The publication can also be passive through harvesting operations from the resource broker, not represented on the picture. The resource broker requests a name service for a unique id (1.1). If the service has any valid GUID, the name service registers the service reachability information and creates a GUID by applying the platform policies for unique naming; otherwise, it registers the service with a link to the original name service (1.2). A valid GUID is sent as response to the resource broker (1.3). The resource broker adds an entry for the service using the GUID and sends a response of successfully publication to the service provider (1.5)

(2) The second task is about the resolution of a GUID. For instance, a service consumer wants to access to a resource, found during a search process (2.0, 2.1, and 2.2). The resource broker uses a name service to resolve the GUID (2.3). The name service may be cascading and contain links to other name services, so that the name resolution may involve different name services. The resolved name is then send back through the request chain until the consumer (2.4), which can access directly the service (3.0 and 3.1).

(3) The last task performs a change of the service descriptions on the name service. The service provider makes a request for service update to the resource broker (4.0). Relevant changes of service information for a name service are version changes, changes on the reachability information, or changes in the naming policy, for instance, when the service provider changes to another domain or service network. The resource broker recognizes that the change request concerns the name service and sends a change request to the name service (4.1). The name service analyses the change requests, updating the service information, e.g. by updating the version number, the reachability entries, or by adding a link to a new name service (4.2). The service GUID that may be changed according to the

name policies is sent back to the catalogue (4.3). The catalogue sends a response back to the service provider (4.4).

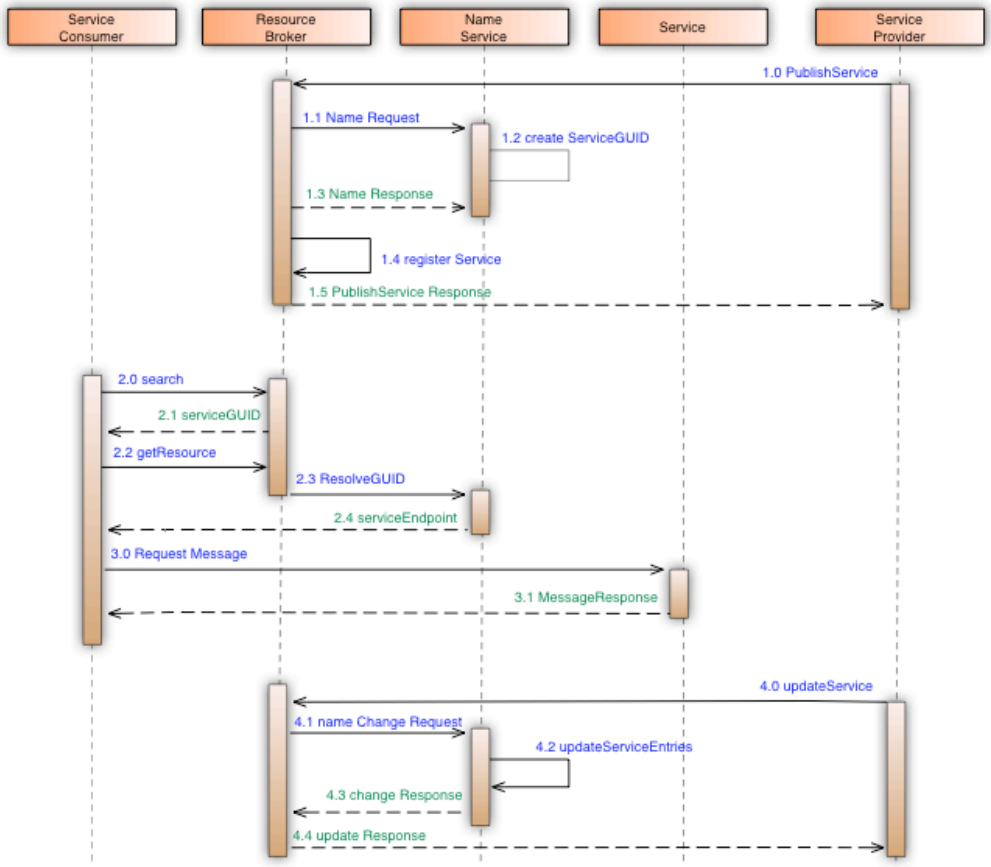


Figure 35: Choreographies for provisioning of unique names

LifeWatch Policy: *LifeWatch will follow the ORCHESTRA approach and specify a Name Service, as a registry and broker providing the following functionality:*

(i) A resource should be registered at a Name Service at the first usage on a service network

(ii) Provision of global unique identifiers at platform level, that means following a particular policy defined in the platform

(iii) Provision of name identifier resolution

(iv) Persistent storage of identifiers, also after the resource is not anymore available

(v) Provision of linkage to all identifiers referring to the same resource instance

(vi) Version management for a resource instance (a changed resource keeps its identifiers, but receives an increased version number, which is available through the Name Service.

7.7.5 Semantic interoperability

7.7.5.1 Semantic goals

Semantics cover a wide field from controlled vocabulary via semantic data integration, knowledge representation, and formal logic to artificial intelligence. In context of LifeWatch, semantic support is envisaged for

- Mediation
- Discovery
- Integration
- Checking consistency

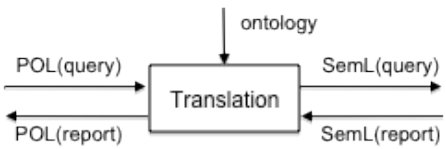
The backbone of semantic support will be provided by domain ontologies (see semantic level of the information viewpoint, Section 5.7) in that, e.g., semantic annotation of resources should refer to these. However, one must acknowledge that neither all users are fluent in the ontological idiom nor annotations of resources conform to that idiom, if annotated at all. Hence, it is necessary clearly to distinguish between those entities that conform to the language and rules of a formal semantics (SemL) and those that do not, but use, e.g., plain (informal) language (POL – plain old language). Automated discovery, data integration, or mediation naturally presumes the existence of formal specification of properties on syntax or semantic level. In contrast, human users will often rely on informal language. Translation of plain language into a semantic query or of a plain text specification into a semantic annotation may be considered as a particular form of mediation service as may be a backward translation of a formal expression into a human-readable format. On the other hand, semantic enhanced services may or should be complemented by services not depending on semantics. For instance, a semantic-based discovery service may be complemented by a discovery service operating similar to web search. One may take even advantage of such a combination in that the latter provides a first set of solutions that is filtered by applying semantic criteria.

7.7.5.2 Mediation

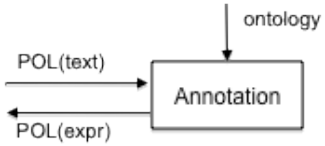
For semantic enhancement of information exchanged in a service network, semantic mediator components can be added at different levels. Semantic mediators provide middleware functionality supporting semantic data access and reasoning for service interactions following different purposes, e.g. discovery, service integration, service interpretation, or consistency checking of models.

Semantic mediator components can serve different purposes the more important ones being listed subsequently.

- Translation – of POL to SemL (typically of simple search queries) and backwards (for reporting)



- Extraction - of information from POL to SemL



- Annotation - storing the semantic information, either as part of the origin entity or in a separate repository with a link to the origin entity

Note: Extraction and annotation are often combined as “annotation service”.

- Mapping
  - Between ontology languages
  - Between ontologies
  - Between data schemas (data types)
  - Between data

Mapping data between different formats may be induced by relating the respective schemata via some ontology and by using data transformation between data and semantic level as well as on the semantic level as in the figure below.

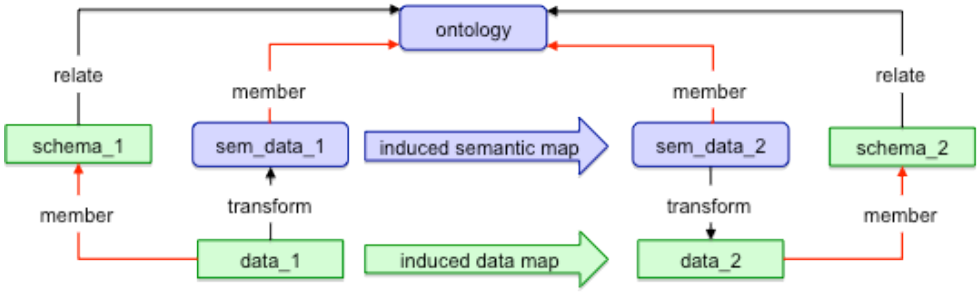


Figure 36: Semantically induced data mapping (version 1)

A further step may additionally use inference between ontologies to achieve the same goal as indicated below.

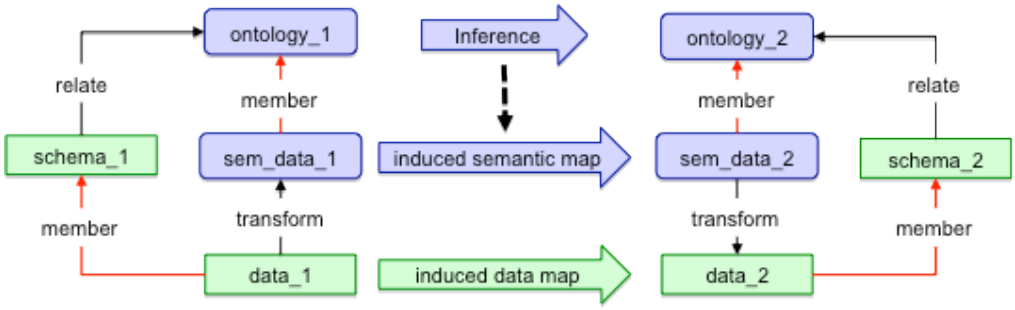


Figure 37: Semantically induced data mapping (version 2)

Further, if the ontologies are represented using different languages, translation between ontologies may be necessary.

- Client-server mediation

Matching service request with service provision may require semantic inference besides syntactic matching, for instance with regard to matching the requester’s goals with the provider’s capabilities as in Figure 38. Semantic enhancement is in particular necessary if matching is to be performed automatically.

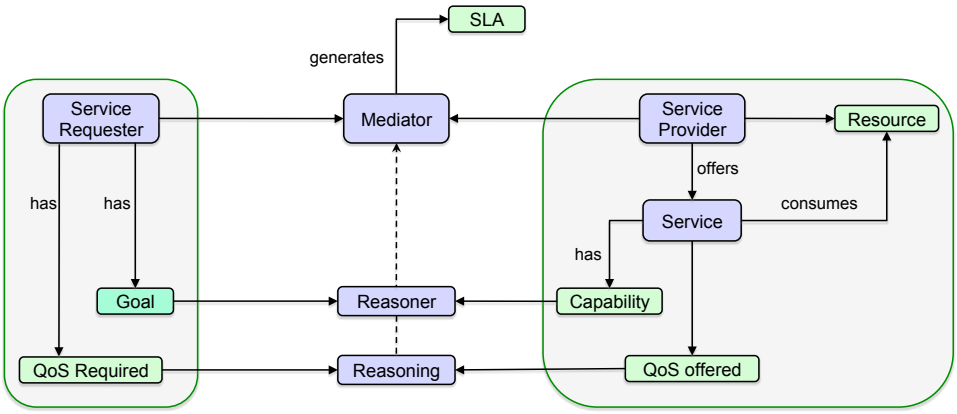


Figure 38: Mediation between service requester and provider

Note: Semantic mediators can be based on ontologies, on data dictionaries, knowledge-based information systems, etc. The discussion above is restricted to the use of ontologies but can correspondingly be applied to other semantic mechanisms.

- Workflow mediation

  The main issue of workflow mediation is the composition of services with different data formats of output and input. Mediator services are that that translate the output of one service to the input of the other, so call shim services. Generating and finding shim services includes the following engineering tasks:

  - Semantic or schematic description of input and output.
  - Generation of a mediation service based on semantic or schematic descriptions of input and output, either automatic or semi-automatic.
  - Automatic retrieval of an appropriate shim service based on semantic description.
  - Building a recommendation system that is based syntactic or semantic descriptions of input and output and that reflects known user preferences.

7.7.5.3 Discovery

A key requirement for service networks is that resources, before they can interact, they have to be visible to each other using appropriate means. Of all relevant application areas for semantic enhancement, discovery probably is the most important one since the discovery component (or resource broker) offers the first "point of contact" between a resource or service consumer and provider by making a resource "findable" and "visible" to potential user searching for solutions to a particular need. There are many kinds of resources such as

- Ontologies
- Data / Data sets
- Services
- Workflows
- Policies
- Contracts

The different resources depend on specific ontologies but use the same general mechanisms. There are several scenarios to consider.

Application scenarios

Publishing

First step is to decide about the ontology (ontologies) to be used or to discover the ontology best suited according to some textual description. Second step is to annotate the resource with semantic information. Semantic information can be handcrafted or automatically generated by an annotation mediator. Finally, the resource has to be registered with a resource broker.

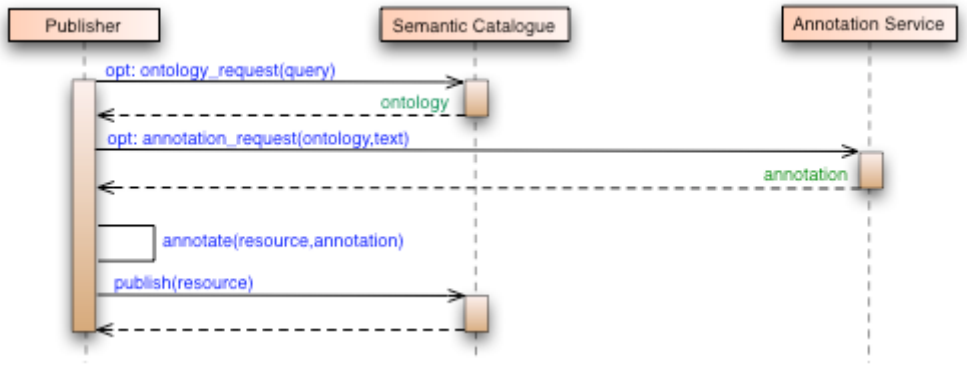


Figure 39: Publishing a semantically annotated resource

Discovery 1

The requester is assumed to be machine agent. The query is a term in some ontology language including a reference to the respective ontology.

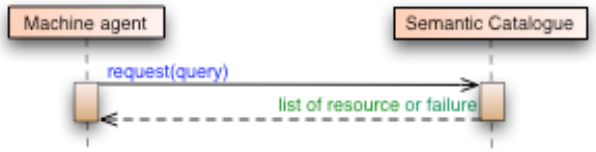


Figure 40: Resource discovery 1

The dialog is trivial but the communication gets more elaborate if the inside of the Semantic Catalogue is considered (see below)

Discovery 2

The requester is a user agent. If the query is a term in an ontology language, behaviour is as above. If the query is a text, there are two alternatives: If the domain ontology is known to which the query relates, a translation request is submitted. Otherwise, some ontology (ontologies) needs to be discovered that relate to the goal (query) and second step again is to translate the query to a semantically valid counter part. The translation may be handcrafted or automatically generated by a translation mediator. Finally, the semantic query is submitted.

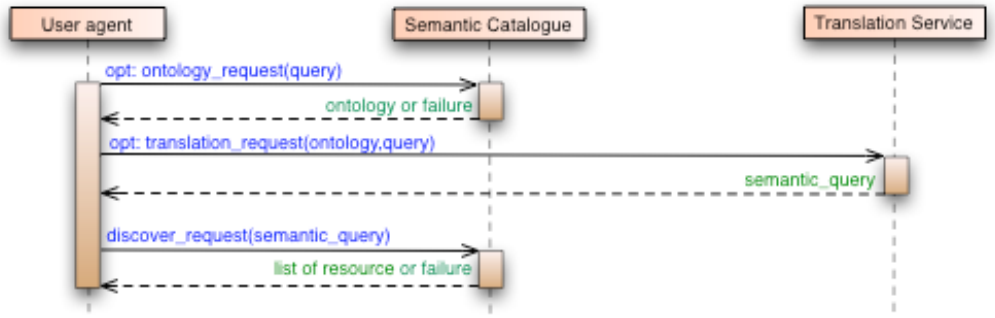


Figure 41: Resource discovery 2

Discovery 3

The requester is a user agent as above. However, discovery starts with a keyword search. The benefit is in the lower complexity of the keyword search since no Inference service is needed (see below). The Inference Service is only applied to the list of resources matching the keyword search.

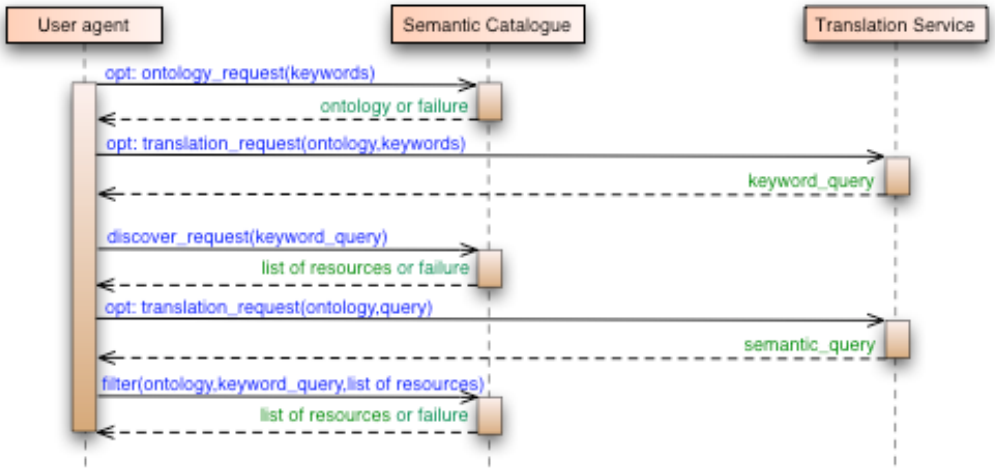


Figure 42: Resource discovery 3

Notes: (i) The "semantic" keyword search may be replaced by a "statistic search" (similar to a classical web search). Semantic search may then be applied to the respective list of result or to the first entries of this list, if it is ordered by "relevance".

(ii) An assumption in all the scenarios above is that the ontologies are expressed in a particular ontology language. Mediation between ontology languages may result in a more complicated picture.

(iii) Discovery of mediation services between different data representation in a workflow (shim services) is considered as a special case of service discovery in that input and output structures are part of the semantic search.

Semantic Catalogue Service internals

The Semantic Catalogue Service should be considered as a front-end to a number of services that cooperate to achieve the desired result. A non-exhaustive list of such services comprises.

- Annotation Service
  Annotation may be automated within the catalogue Service. Then for instance, different to the Publishing Scenario, the publisher would not explicitly influence the annotation process

- Inference Service (Reasoning Service)
  Inference may be needed to match a query with the capabilities offered. Inference is costly in computational terms; hence, semantic discovery strategies should try to minimize the application of inference.

- Ontology Access Service – Ontology Discovery Service
  Ontologies must be stored and accessed. Particular discovery strategies should be supplied that allow finding ontologies that relate to specific keywords or text. In that, the Ontology Discovery Service is a particular instance of a (resource) Discovery Service.

- Query Mediation Service
  Many resources will be only accessible by conventional catalogues. A Query Mediation Service relates the semantic and the conventional catalogues. Given a semantic query the Query Mediation Service generates adequate sub-queries trying to match the semantic query with the meta-information entries in the conventional catalogues. If resources are found, the Query Mediation Service either retrieves the resource or assembles a report possibly using an annotation service on the fly. Given that the results conform to the user's requirement, the reference to the resource enhanced by the semantic annotation may be added to the semantic catalogue and/or, if the original resource allows for semantic annotations, the semantic annotation may be added to the original resource.

Discovery with regard to specific items

- Ontologies
  Ontologies may be discovered using keywords and a general search engine with additional information about the kind of documents looked for (e.g. by extension .owl, .rdf). A specific search engine like Swoogle (http://www.swoogle.de/) may be more focussed. The alternative is to browse ontology registries or libraries. These are usually domain-specific. Keyword search may be enhanced by more semantic queries, for instance, by querying for the existence of certain triples. Incremental filtering mechanisms may be used to cut down the number of ontologies, but the final decision which ontologies to use usually depends on human interaction.

- Services
  Semantic descriptions often depend on the nature of input and output data, enhanced by functionality information. Quality of service and other non-functional properties may as well be provided.

- Workflows
  Gil et al. [Gil et al 2009] state that scientists discover workflows given properties of workflow data inputs, intermediate data products, and data results. They provide an approach to workflow matching based on:

    - An algorithm for enriching workflow catalogues with relevant properties of data and components,
    - An algorithm for workflow matching based on data centred properties, and
    - Separation of reasoning steps to be called outside the data catalogue.
    - Derivate information from existing information

7.7.5.4 Integration

LifeWatch has to be aware that the meaning of concepts, no matter whether those are concepts within controlled vocabularies of concepts of metadata or concepts behind the generation of data or concepts for

service descriptions and interface descriptions will differ. Although standardizations are desirable, they cannot be applied retroactively so that historical data tend to follow divergent semantics. Moreover, science is always in development thus producing new concepts. Therefore, one of the goals of LifeWatch will be to integrate those divergent and heterogeneous conceptual worlds.

Integration may take place on several layers:

- Ontology
  Integration of ontologies relates to the concept of modularisation and composition of ontologies. The subject of modularisation and composition is a long-standing subject with regard to general logical theories as well as for application areas in computer science such as Abstract Data types. The inherent problem is to find mechanisms that are conservative in that properties that hold for the individual should be persistent in the composition, meaning that the properties that hold for the individual module should hold in the composition but no additional properties should hold for the (then) sub-module. This is inherently difficult since deduction may generate additional properties to hold or even inconsistencies. (For an overview about modularisation of ontologies see: [Stuckenschmidt e al.]89, also: International Workshop on Ontologies: Reasoning and Modularity90)

- Schema
  As with ontologies, integration of data schemes essentially is a question of modularisation. However, schema composition is much simpler since no deduction is involved. The main issue is the proper separation of concerns with regard to the sub-modules used.

- Data
  The most common scenario of data integration is that data are distributed across several databases and that a (semantic) query for data is issued where the response data is drawn from several of the databases. The prevailing engineering solution seems to consists of two main components:
  - A wrapper component for each database that relates semantic language to the language used for data representation and query in the particular data base,
  - A mediator component that transposes the original query into a query plan, i.e. a series of sub queries to be submitted to the relevant databases. The planner depends on contextual information provided by the wrapper components in order to direct the sub queries to the appropriate database.

7.7.5.5 Consistency

Consistency means absence on contradiction.

- Ontology
  Inconsistency for a ontology implies that both a statement and its negation can be proved, hence renders an ontology useless.
- Data
  Data may hold inconsistent information. An inconsistency may be due to syntax errors using the wrong format or, e.g., presenting a date in different formats as strings, using values that are out of boundaries, using different scales. Syntactic and semantic checks are to be devised to avoid such

---

89 Heiner Stuckenschmidt, Christine Parent, Stefano Spaccapietra (Eds.), Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularisation (Lecture Notes in Computer Science / Theoretical Computer Science) von

90 https://dkm.fbk.eu/worm08/

inconsistencies. These may be generated (semi-) automatically starting with a more sophisticated type check on the syntactical side.

- Data sets
  Data sets can contain inconsistent or incomplete data. Data cleansing should be applied to obtain consistent data of the same degree of completeness. In case of uncertain or incomplete information, the degree of uncertainty should be marked rather than to reject data. A strategy needs to be defined to add uncertainty information. The degree of uncertainty should a priori be part of the semantic description of data and, if not yet inherent part of source data, algorithmically added in the data integration phase.
  A second kind of inconsistency in a data set may occur if different data items are conflicting. Resolution mechanisms are to be defined that resolve such conflicts based on the given semantic descriptions.

In any case, one cannot expect that conflicts can be resolved automatically. However, the algorithms used should at least expose found conflicts for human inspection.

7.7.5.6 Methodological aspects

The methodological approach should combine a top-down with a bottom-up approach, including the following steps:

1. Creation of the basic structure for the integration
   a. Definition of a core ontology (adopt and adapt an existing one)
   b. Design a meta-thesaurus
   c. Create a meta-thesaurus
   d. Maintenance of meta-thesaurus
2. Integration of controlled vocabulary:
   a. Definition of service for controlled vocabulary (protocol and format)
   b. Implementation of an access service for controlled vocabulary
   c. Definition of best practice for the establishment of new controlled vocabulary
3. Semantic mediation and mapping (semi automatic maintenance work)
   a. Semi automatic process for establishment of semantics out of unstructured data and structured data
   b. Definition of services for semantic annotation of datasets and data
   c. Creation of services for semantic annotation of datasets and data
   d. Definition of semantically annotated interfaces for services
   e. Semantic mapping of workflows

7.7.5.7 Policies

At the present stage, it is difficult to decide upon policies due to the fluent state of many of the questions related to semantic interoperability. Hence, only some general policies can be stated. The arguments and descriptions above give some hints which additional policies may be useful in future.

LifeWatch has to be aware that the meaning of concepts will differ and develop, no matter whether those are concepts within controlled vocabularies, concepts of metadata, concepts behind the generation of data, or concepts for service and interface descriptions. Although standardizations are desirable, they cannot be retroactively applied so that historical data tend to follow diverging semantics. Moreover, science is always in development thus producing new concepts, requiring adoption and integration of newly emerging, diverging, and heterogeneous conceptual worlds.

LifeWatch Policy: (i) LifeWatch shall support semantic interoperability providing as much flexibility as feasible to integrate new developments but shall adopt standards whenever these emerge.

(ii) LifeWatch shall provide the following non-exhaustive list of semantically enhanced services (or a multiplicity of these):

- *Annotation Service*
- *Catalogue Service*
- *Inference Service*
- *Ontology Access Service – Ontology Discovery Service*
- *Query Mediation Service*
- *Translation Service*

(iii) Semantic interoperability is supported by meta-information models for the purposes "Discovery", "Integration", "Mediation", and "Orchestration". These meta-information models shall be defined and extended simultaneously with the development of the semantic framework of LifeWatch.

(iv) Integration mechanisms used shall be persistent in that the original content of the components can be retrieved.

7.7.6 Controlled access to resources

Right-managed access deals with the question of whether a particular request for a resource will result in that resource actually being returned. In LifeWatch, Authentication, Authorisation, and Accounting Services will perform access control. Because of the federated nature of LifeWatch, these services will be federated. Authentication refers to the process by which one verifies that someone is who (s)he claims (s)he is. Authorization is finding out if the person, once identified, is permitted to have the resource. Accounting refers to the tracking of the consumption of network resources by users. Access control may go beyond in access can be granted or denied based on a wider variety of criteria, such as the network address of the client, or the browser which the visitor is using, it's controlling entrance by some arbitrary condition which may or may not have anything to do with the attributes of the particular visitor.

Federated identity

Federated authentication is often discussed under the heading of Federated Identity. Federated identity management refers to a set of technologies and processes that let computer systems dynamically delegate identity tasks across different security domains, for instance in terms of single-sign-on, which let users authenticate only once to get access to protected resources in different security domains. Besides dealing with security risks as the main objective, federated identity management involves privacy risks in that information may be shared across domains otherwise uncoupled. Minimisation of both risks poses new architectural and engineering challenges.

A federated identity model involves four agents91:

- The user as the person who assumes a particular digital identity.
- The user agent (UA) as the application via which the user interacts with the digital network.
- The service (resource) provider (SP) who provides resources.
- The identity provider (IdP) where the user logs in.

In general, each organization participating in a federation operates one Identity Provider for their users and any number of Service Providers (see Figure 43).

---

[91] E. Maler, D. Reed, The Venn of Identity: Options and issues in federated identity management, IEEE Security & Privacy, http://doi.ieeecomputersociety.org/10.11

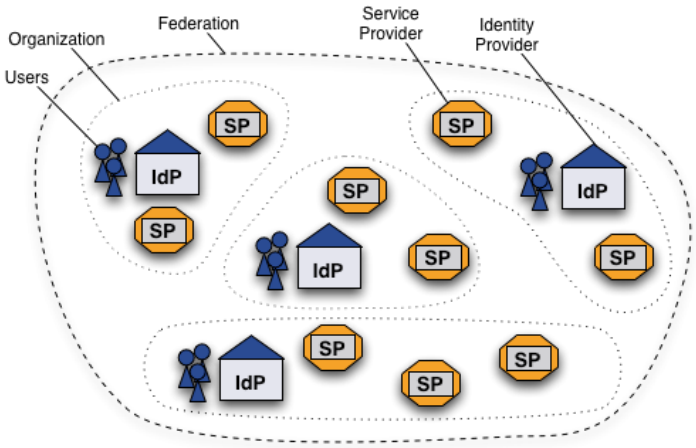


Figure 43: Federation92

There are several patterns for implementing single-sign-on that may be distinguished by being SP-initiated or IdP-initiated. In the SP-initiated case, the SP has send explicit authentication requests to the IdP. In the IdP-initiated case, SPs are accessed via the IdP, for instance via a portal. An example for an SP-initiated choreography is a Shibboleth93-like login procedure (see Figure 44).

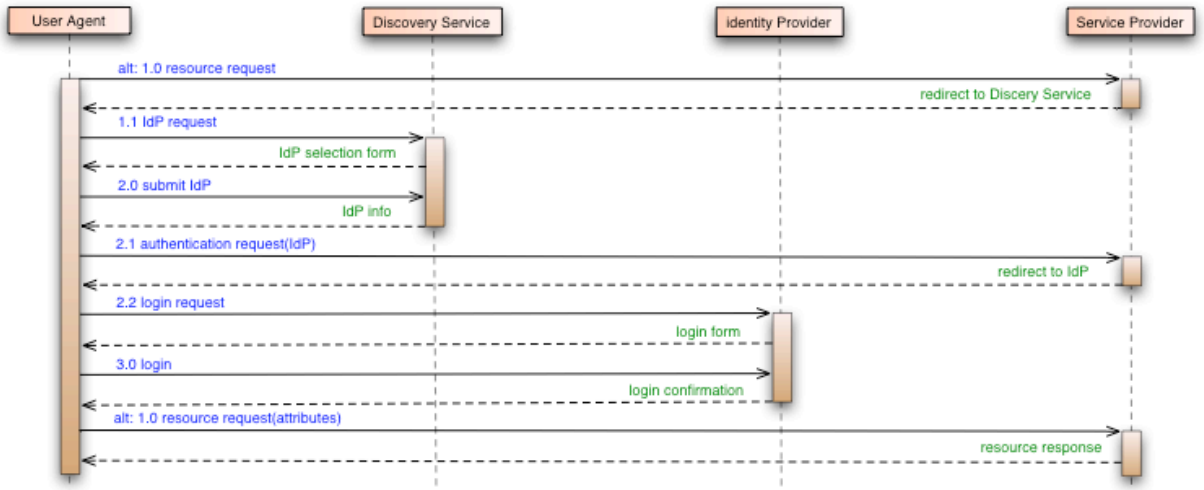


Figure 44: Shibboleth login procedure

There are three phases:

- The user issues a resource request. If the user is logged in, the second alternative applies: the user supplies credentials and is given access provided the credentials are sufficient. Otherwise, the response is a request for IdP information that is redirected via the user agent to a "Discovery Service" that offers a form where the user can choose the appropriate identity provider.
- The user submits IdP information. The response triggers an authentication request for the chosen identity provider that is submitted through the user agent. The identity provider evaluates the authentication request and answers with a login form.

92 Source: http://switch.ch/aai/demo/easy.html
93 http://shibboleth.internet2.edu/

- The user provides its credentials to the identity provider and an assertion including the user's attributes is created. The attributes are attached to the original resource request. If these are sufficient, the resource is accessed.

There is a variation to the request-response where SP is provided with a handle only and acquires the credentials directly from the identity provider.

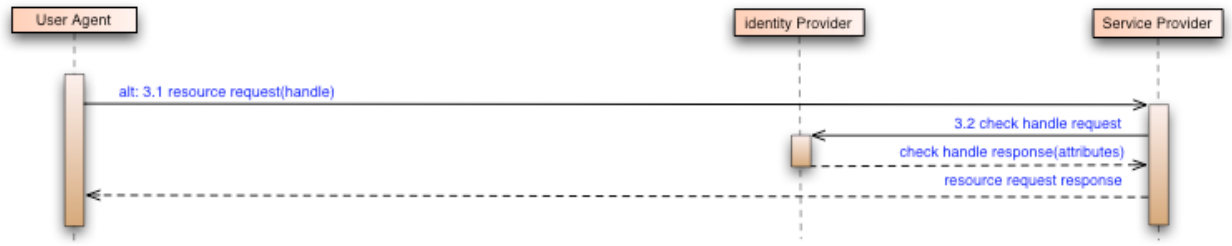


Figure 45: Variation of the login procedure

Note that all communications are stateless.

As such, sequence diagrams only provide sequences of messages to be followed. They do not provide information about the responsibilities and only vaguely about the content of messages. In particular, it is not manifest who holds which information at which point of time. For instance, the final resource request requires the user attributes as provided by the IdP. The attributes should be filtered to provide only the necessary credentials for the particular resource required, but the user's attributes may contain information about many resources allocated to many service providers the user is allowed to access. Hence, a filtering process is needed according to resource: the user agent must submit not only the login credentials but also the resource information in step 3.0. Whether the latter information is inherent to the user agent or whether it is transmitted all the way along from the initial resource request needs to be specified as an inherent part of the specification of the choreography for federated identity.

Another matter of concern is the analysis of risks involved. Though convenient for the user and from a system point of view – the user logs in only once, the service provider is not bothered with managing user account, the identity provider can focus on improving authentication methods – new security and privacy risks are generated. Crossing security domains enlarges the security risk surface. Securing the communication channels against malicious attacks (by, e.g., SSL/TLS[94]) is a minimal requirement. However, if many organisations are federated, the risk of some gap in the security framework increases.

On the other hand, privacy risk increases by using a unique identity. For instance, given that the identity is revealed, by tracing the activities of a scientist one may become aware of the research agenda and possibly research results. Using encrypted, temporary identities only linkable to the real identity by the identity provider may here be a solution if, for instance, the credentials do not depend on the individual user. If they do – a particular use is, for instance, allowed to access the location information of rare species – another problem occurs; service provider and identity provider have to agree on credentials of the particular user and the service provider must be able to check that the individual really is the one it pretends to be. For instance, they service provider may generate an encrypted attachment to the user credentials that grants access to the resource when decrypted by the service provider. Then the global identity of the user does not need to be revealed if accessing the resource.

The upshot is that authentication schemes for a federated identity needs to be carefully crafted balancing between ease of use and minimisation of risk

[94] Secure Sockets Layer / transport Layer Security

LifeWatch Policy: *(i) Within the construction phase, LifeWatch shall define some choreography for dealing with federated identity.*

(ii) The federated identity scheme should minimize security and privacy risks.

(iii) The authorisation scheme should be uniform for all LifeWatch Service Networks.

(iv) The choreography outlined above should be considered.

7.7.7 Provenance

Objectives of provenance in LifeWatch are (see Section 4.3.6):

- Estimation of data quality and reliability
- Logging of Audit Trails
- Replication of data and in-silico experiments
- Attribution of copyright and ownership of data
- Informational for discovery

Two branches of provenance research can be distinguished between which little interaction seems to take place but both of which are necessary to develop a full understanding of the provenance of data.

- For data-oriented provenance, the derived data are the primary entities for which provenance is collected. For example, a graph of transformation steps is directly associated with the data item being derived without reference to an external mechanism.
- For process-oriented provenance (or workflow provenance), the deriving process is the primary entity for which provenance is collected, and the data provenance is determined by inspecting the inputs and outputs of these processes.

Workflow provenance, when interacting with a database, seems in general only to record the state of a database and the query, which assumes some form of persistence or archiving.

The present state of research makes it difficult to state solid engineering principles on which provenance in LifeWatch may be based, except for some general guidelines. These engineering principles need updating as the state of art proceeds.

LifeWatch Policy: *(i) LifeWatch provenance information shall at least support attribution (ownership and copyright).*

(ii) LifeWatch provenance will use workflow provenance as presently supplied by workflow systems.

(iii) LifeWatch provenance shall be flexible to handle the different formats presently in use.

(iv) LifeWatch will adopt and contribute to emerging provenance standards.

7.7.8 Workflows for Orchestration

The major objectives of the LifeWatch workflow environment are:

- Support for the definition of workflows.
- Management of workflows as reusable functional components, in particular, workflows maybe considered as services to be used as elementary building blocks in new workflows.
- Mapping from the workflow to the underlying resources.
- Enactment of workflows on the underlying resources.

- Provision and maintenance of the metadata descriptions of services and workflows.
- Bookkeeping of the results and generation of provenance information.
- Sharing of workflows.

These components may be integrated in one tool. Separation of the two phases of the workflow authoring and the enactment, however, offers a number of advantages:

- Remote machines that do not consume the resources of the user's workstation can execute the workflow.
- The workflow can be executed at a time different from that of its creation.
- Once created, the workflow can be subsequently modified using a different workflow-editing tool (i.e., by another user), without affecting the ability to enact the workflow.

Communication between the two components should be unidirectional in that only the client makes requests to the server side with the advantage of that being firewall friendly.

Interaction of authoring and enactment depends on an interface language for the specification of workflows. Unfortunately, a standard for a workflow description language is an open issue. For business processes, BPEL and XPDL seem to be quasi standards. The general understanding, however, is that these languages are not particularly well suited to the needs of scientific workflows. However, no (quasi-) standard workflow definition language has emerged at the present stage though its existence would greatly facilitate the development of and interchange between workflow tools.

LifeWatch Policy: (i) Workflow authoring and workflow enactment shall be separated.

(ii) If a standard for a workflow definition language is emerging. LifeWatch shall adopt the standard and support the development of such a standard.

There are a number of further requirements that at this stage are not yet defined as policies but should be considered as recommendations:

- A modular design that, on the basis of basic editing and enacting facilities, allows enhancement by, e.g., plug-ins or toolboxes;
- Navigable presentation of the available services and workflows;
- Graphical composition and linking of services based on the descriptions of their inputs and outputs;
- Support of hybrid workflows;
- Support of different levels of granularity, combining, e.g., "coarse-grained" web-service based workflows with "fine-grained" computational components encapsulating algorithm obtained by visual programming or other programming paradigms;
- An intuitive GUI should support the user to compose a workflow visually from smaller components, or to "drill-down" into sub workflows, to animate workflow execution, or to inspect intermediate results;
- Syntactic analysis of services and provides feedback to the user for the compatibility of two services when connected together;
- Semantic validation of the workflows assuming that the participating services have been annotated semantically;
- Support the persistence and retrieval of workflows;
- Support for user-interaction: Many scientific workflow srequire user decisions and interactions at various steps. Using a notification mechanism the user might be asked to reconnect to the running instance and make a decision before the paused (sub-) workflow can resume;
- "Smart" re-runs after user-interaction: only those parts of a workflow are re-executed that are affected by the user interaction;
- Versioning support for workflows;

- Interaction with a Metadata Repository that, for instance, provides the listing of the available services and the syntactic and semantic validation of workflows;
- Support of "smart" semantic links: When designing a workflow, assistance may be given to the user by suggesting which component might be appropriate in the particular context. Such assistance may be particularly useful with regard to so-called shim services (data format translation services);
- Interaction with Grid Data Management services: workflows may involve large volumes of data and/or require high-end computational resources. Such data-intensive and compute-intensive workflows must be supported by suitable interfaces to Grid middleware components (sometimes called Compute-Grid and Data-Grid, respectively);
- Capabilities to control and monitor the execution of the workflow and its individual services, allowing to stop a workflow or to navigate through the process steps, e.g. repeating the previous step or, if possible, skipping a step or changing parameters during the execution;
- Provision of strategies for improving reliability and fault-tolerance, e.g. to look for alternative services when a service becomes unacceptably slow or are unavailable;
- Minimisation of communication between client and server and keep as much as possible business logic in the client in order to avoid the inherent latency of the network which may negatively affect the user's experience;
- Support of the access policies and security constraints of LifeWatch, e.g., allowing single-sign-on for all resources used in a workflow;
- To enhance flexibility workflows should be considered (and deployed) as "higher order" composite services.

8 Technology viewpoint

8.1 Introduction

The technology viewpoint specifies the technological options for platforms, their characteristics, and their operational issues. The design of a LifeWatch Service Network comprises the specification of a service platform and its characteristics onto which the LifeWatch Services and Application Schemes are to be mapped. The requirements for the specification of service platforms follow those of ORCHESTRA. They are listed in Section 8.2 together with possible choices for a web service platform.

The technological options depend on the current technologies. They influence service design and implementation even on quite an abstract level.

On one hand, creation of information models for services depends on how the structures provided by a service are defined, if they are, e.g., rather atomic information elements, also called resources, having a common interface or if they are considered functional entities supporting operations on a set of atomic information elements. This aspect influences the interface languages and the protocols used. The general design approach for the service platform should be indicated in the specification, although the platform may as well support different design paradigms (e.g., RESTful service and SOAP services) through intermediate services. If so, this should also be indicated in the specification. Section 8.2 will discuss several of these options.

On the other hand, the way a service communicates with a service consumer and with other services in order to perform a particular activity is determined by the distribution architecture. Section 0 outlines current technologies for distributed systems.

The technology viewpoint concludes by presenting technological options for specific capabilities of the LifeWatch ICT infrastructure in Section 0.

The recommendations of this chapter are based on the complimentary status report on Infrastructures for Biodiversity Research where descriptions of all the concepts, technologies, and components referred to can be found. Hence, no explicit references are provided.

The following general strategy should be adhered to for technologies:

LifeWatch Policy: *(i) LifeWatch technology shall exploit available standards whenever appropriate and feasible. This includes quasi-standards build on best practice. But even standards should be used with care since standards may be competing reflecting particular interests and they may not be fully accepted by the community.*

(ii) Whenever possible, development should be based on and integrated with open source to involve the community and thus to enhance sustainability.

8.2 Service platform

8.2.1 Requirements for the specification

A service platform specification is a realisation of the meta-class OMM_PlatformSpec (see Appendix C). The components of a platform specification are:

- Platform Name
- Reference Model

If the platform specification is based on a specific version of the LifeWatch Reference Model, the version number shall be provided

- Interface Language
  Formal language used specification of service interfaces
- Execution Context
  Specification of the Execution Context as an agreement between service providers and consumers that contains information about, e.g., preferred protocols, semantics, policies, and other conditions and assumptions that describe how a service can and may be used.
- Schema Language
  Specification of the schema language for defining Information Models
- Schema Mapping
  Specification of how to map platform neutral information model to the schema language used for the particular platform.
- Information Model Constraints
  Specification of the constraints on the LifeWatch Information Model, especially the constraints on the message format required for accomplishing the LifeWatch Action model.

Specific aspects that may be considered at the platform level are:

- User Management, Authentication, and Authorisation
  The LifeWatch Reference Model provides a platform neutral specification of concepts for User Management, Authentication, and Authorisation (UAA) in order to be able to cope with established UAA mechanisms for dedicated platforms. For a specific platform, the platform neutral specification may have to be refined to agree with the authentication and authorisation mechanisms provided by the specific platform.
- Data Formats
  An agreement on the usage of standard data formats and specific/proprietary data formats may be part of platform specification.
- Platform Mapping
  In case that information models are modelled directly in a platform-specific schema language, conformance of such information models to the LifeWatch Meta-model has to be ensured: it must be possible to generate the UML representation out of a platform-specific specification such that the mapping rules for Application Schemes generate the original platform specific specification.
- Mapping of Service Interfaces
  Procedures for the mapping of the platform-neutral service interfaces to a specific interface language may have to be defined. These procedures shall ensure that the mapping is in compliance with the rules of the LifeWatch Service Meta-Model. The mapping itself shall be part of an implementation specification. Ideally, the mapping should be bi-directional in that the platform neutral interface specification can be automatically retrieved.
- Restrictions on Certain Services
  A platform specification may reduce the complexity or restrict the scope of certain services, if this is required to meet the main characteristics of the selected platform. This may, for instance, be the case if a platform provides semantically similar services of restricted nature. Note that this complicates interoperability between different platforms.

A platform specification may include recommendations for libraries and tools.

8.2.2 Platform example: components of a web service platform

Possible choices for the definition of a web service platform are sketched.

Interface language

Interface language may be a Web Service Description Language such as WSDL, or SAWSDL if semantic information is incorporated. WADL may be the choice in case of a RESTful web service. The precise version number is part of a platform specification.

Execution context

The message exchange protocol may be, for instance, SOAP 1.2 with HTTP as transport protocol. The message style may be defined in terms of a MIME type, for example “document/literal non-wrapped”.

For security, the basic mechanism may be SOAP Message Security for encryption of SOAP messages. As a requirement, the execution context may state that session information must be included in the SOAP header. RESTful services might require HTTP Basic or HTTP Digest or SSL certificate-based mutual authentication for propagating identity credentials.

The format of machine-readable descriptions of services may be another topic to include in the execution context. In general, whatever is used to run services should be included here with references to all the relevant documents, version information included.

Schema language

An example for a schema language for the geospatial domain would be the Geography Markup Language (GML) or the Ecological Markup Language (EML) in case of application in the ecological domain. Of course, more than one schema language may be referred to.

Schema Mapping

A schema mapping can be defined in terms of encoding rules. For instance, given that the schema language is GML, the rules of ISO 19136:2005 Geographic Information – Geographic Markup Language (GML): Annex E: UML-to-GML Application Schema Encoding Rules.95 These schema encoding rules are based on the general idea that the class definitions in the UML application schema are mapped to type and element declarations in XML Schema, so that the objects in the instance model can be mapped to corresponding element structures in the XML document as defined by the following general constraints on conversion rules.

UML application schema	GML application schema
Package	One XML Schema document per package (default mapping)
<<Application Schema>>	XML Schema document
<<DataType>>	complexType, property type and global element
<<Enumeration>>	Restriction of xsd:string with enumeration values
<<CodeList>>	Union of an enumeration and a pattern (default mapping, an alternative mapping is a reference to a dictionary)
<<Union>>	Choice group whose members are GML objects or features, or objects corresponding to DataTypes
<<FeatureType>>	Global element, whose content model is a globally scoped XML

[95] http://portal.opengeospatial.org/modules/admin/license_agreement.php?suppressHeaders=0&access_license_id=3&target=http://portal.opengeospatial.org/files/index.php?artifact_id=20509

	Schema type derived by direct/indirect extension of gml:AbstractFeatureType, property type
No stereotype or <<Type>>	Global element, whose content model is a globally scoped XML Schema type derived by direct/indirect extension of gml:AbstractGMLType, property type
Operations	Not encoded
Attribute	local xsd:element, the type is either a property type (if the type is a complex type) or a simple type.
Association role	local xsd:element, the type is always a property type (only named and navigable roles)
General OCL constraints	Not encoded

Information Model Constraints

For instance, the ORCHESTRA web service platform requires that, to avoid technical complications, all references GML in a WSDL interface shall be substituted by the XML schema base type xsd:string.

8.3 Technology options for services

The present version of the technology viewpoint assumes that the implementation of LifeWatch services will be based on web services technologies where "web services" are meant as an abstraction and do not refer to a particular technology e.g. the so called Big Web Services or WS-* standards. The W3C definition of Web Service can be relaxed by suppressing the technology aspects, since they are, in LifeWatch terms, platform specifications on descriptions, interfaces, and schema languages.

The LifeWatch Reference Model will use the following modification of the W3C definition of web services "a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processible format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP-messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards."96

The W3C technical report Web Service Architecture97 presents four models that focus on different aspects of the architecture: the message oriented model, the service oriented model, the resource oriented model, and the policy-oriented model. These models not only focus on different aspects but define different design paradigms for modelling of application components and information schemas. This holds in particular for Message Oriented Architectures (MOA), Service Oriented Architectures (SOA), and Resource-Oriented-Architectures (ROA).

[96] W3C definition of Web Services (without the crossed out parts) available at http://www.w3.org/TR/ws-gloss/

[97] W3C (2004): Web Services Architecture, W3C Working Group Note 11 February 2004, available at http://www.w3.org/TR/ws-arch/ (November 2009)

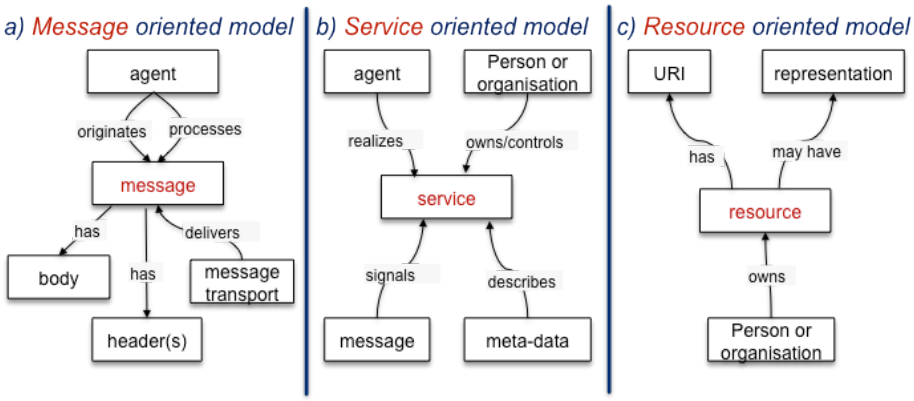


Figure 46 Simplified models used on web services design, according to W3C98

8.3.1 Message Oriented Architecture (MOA)

MOA focus on asynchronous interaction for loosely coupled and scalable "event-driven" systems. The design models the data dissemination as sequences of "messages" passing among message handlers, without relating a service to a particular service. Figure 46 a) shows a simplified model of the concepts used on a message oriented model according to W3C: an agent (computational resource) sends and receives messages structured in terms of message headers and bodies over a delivery mechanism (message transport). Ubiquitous computing and sensor technologies are examples of systems relying on MOA. Also the software architecture pattern Event-driven architecture (EDA) is based on MOA.

Relevant technologies

Editor's Note: *To Be Completed.*

8.3.2 Service Oriented Architecture (SOA)

Classical SOA is based on services executing actions (operations). Therefore, also the term activity-oriented may be applied for this type of architecture. A service interface hides the data structure behind the actions and offers only operations, each of one executing one of the activities or actions that can be performed. Figure 46 b) shows the concepts of the W3C simplified model: an agent, also called service provider realizes services that are used by another agent, called service consumer. Services are mediated by means of message exchanges and are described through meta-data. The use of (standardized) meta-data is a key for the successfully deployment, discovery, and access of services. SOAP is a protocol for service-oriented architectures using HTTP as transport mechanism, tunnelling all messages in envelopes sent using HTTP-Post requests. SOAP and the W3C Web-Service protocol stack (also called Big Services or WS-*) together with RPC-styled service architectures are the predominant examples of the SOA architecture paradigm.

Relevant technologies

Editor's Note: *To Be Completed.*

[98] Images adapted from: W3C (2004): Web Services Architecture, http://www.w3.org/TR/ws-arch/ (November 2009)

8.3.3 Resource Oriented Architecture (ROA)

ROA has its origins in a specific set of architectural constraints named REST or Representational State Transfer by Roy Thomas Fielding99. REST exemplifies how the web's design emerged. REST-based architectures basic concepts is the transfer of representation of resources, that is, according to W3C, "anything that can have an identifier" and are relevant to web services. Figure 46 c) shows the simplified W3C model of ROA that focuses on resources being identified through an URI and having none, one, or many representations and an owner100. An important component missing in the model is the link concept representing relationships between resources. The term ROA is used to describe architectures for web services based on the REST principles, also called RESTful architectures101 and the distinguishing properties are addressability, connectedness, and the uniform interface more attached to database-oriented operations. ROA, in contrast to SOA, uses HTTP as message protocol and not just as transport mechanism. The "Status Report on Infrastructures for Biodiversity Research" (Deliverable 5.1.4), Appendix B, provides additional material on comparing SOA and ROA.

While the engineering viewpoint provides the concepts and guidelines for the architectural design of LifeWatch applications based on SOA it is worth to keep in mind that technological decisions may influence the design of application schemas. To illustrate Figure 47 represents two design examples (SOA and ROA) for the ORCHESTRA architectural service User Management Service that controls the setup of user structures in order to be used by the authentication and authorisation mechanisms. The schema shown in a) matches the SOA-view as described by the ORCHESTRA-Reference Model102 and in b) a ROA-design is presented.

---

99 Fielding, R., T., Architectural Styles and the Design of Network-based Software Architectures, Ph.D. Thesis, University of California, Irvine, Irvine, California, 2000.

100 The WRC Web Service Architecture expands the definition of resource specified in the Web Architecture document (W3C, 2004: Architecture of the World Wide Web, Recommendation 15 December 2004, http://www.w3.org/TR/webarch/) by incorporating the relationship between resources and owners.

101 The term Resource Oriented Architecture (ROA) was formulated in Richardson, Leonard; Sam Ruby (May 2007). RESTful Web Services. O'Reilly.

102 The interface ServiceCapabilities was omitted in the representation

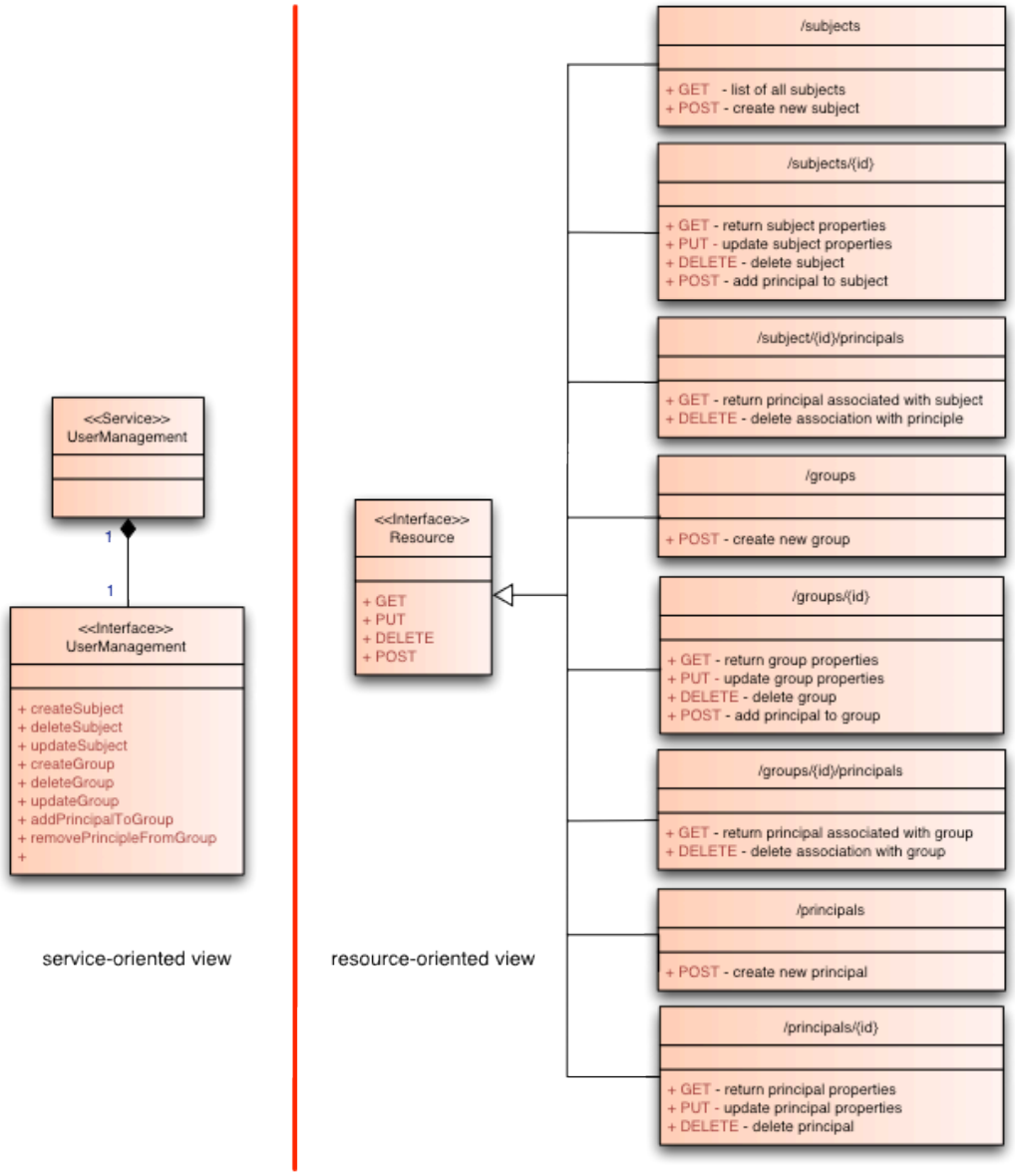


Figure 47: Warehouse application design as SOA (a) and ROA (b)

The SOA model (a) has just one interface, and thus one address to with client can call a long set of operations. Looking more detailed to the operations, they are restricted to creation, read, update and delete operations, often called C.R.U.D. operations, on different data types (subjects, groups, and principals), which are passed as parameters for the operations. In contrast, the ROA model (b) identifies the key resources and apply a set or subset of operations as defined by the resource interface, which are few, namely get, put, post, and delete. Clients can access each resource using a unique address (URI), for example using HTTP get on http://{service_base_URL}/subject/JohnDoe returns the entries for a subject with the ID JohnDoe. On a SOAP implementation of the SOA model the operations are mostly[103] performed using the HTTP-POST operation and hiding the semantic of the query and the data types in an envelope. The ROA implementation would use HTTP-POST just for adding children to a resource, e.g.

[103] Since SOAP 1.2 also the operator HTTP-GET is supported

using POST on subjects creates a subject and returns an URI that can be used to update or delete the specific subject using HTTP-PUT and HTTP-DELETE on the URI respectively104.

Relevant technologies

Editor's Note: To Be Completed.

8.4 Technology options for platform architectures

Service oriented architecture is a generic paradigm for sharing resources and data in a network, in other words, it complies with the distributed computing goal of dividing a problem into many tasks, each of which is solved by one computer. Distribute computing can be seen as a loosely form of system components running concurrently in parallel105 and have the following characteristics meeting the requirements for the ICT infrastructure listed in section 4.1 (and as well expanded in Section 7.2 as engineering guidelines):

- Resource sharing: Resources (hardware, software, or data) are physically encapsulated within one of the computers and can be accessed from other only by communication.
- Concurrency: Concurrency arises from the separate activities requested, the independence of resources, and the distributed location of computational processes, thus different processes may access the same resource concurrently.
- Scalability: Distributed systems are easy to increase in scale, ideally without changing existing components. Examples of scaling factors are adding more processors, resources, or users while accommodating the corresponding responsiveness of the system.
- Fault tolerance: Distributed systems should be reliable and ensure the operation without degrading significantly performance and functionality in case of failure of one or more resources. Recovery of software and data and redundancy of hardware are mechanisms to achieve it.
- Transparency: Distributes systems should hide their complexity to the users and application programmers, presenting an integrating whole to facilitate usage and implementation.

There are different paradigms of architectures to realize distributed computing. The next subsections will outline them with the current technologies as a basis to be considered when choosing and implementing a particular platform.

8.4.1 Client-server architecture

The client-server architecture is a model for the basic SOA principles: a client (service requestor) communicates with a server (service provider) via messages exchange.

There are different approaches to layer client server architectures, in dependence on the number of logical components. In typical client-server architectures, also called two-tier architecture, the server holds and process the data (e.g. a remote database or a web server) while the client contents the application logic. This architecture can be extended to a middle tier (also called middleware or application server) that holds the application logic, separating it from the data tier (at the server) and from the presentation tier (at the client). Most of the web applications follow this model. When many applications use the same services, then the application logic can be divided into different layers, e.g. for data access, for security issues, for particular processing services, and for presentations. This is the basic idea behind (enterprise) application servers or network applications, which separates (business) logic and (business) processes to

104 A detailed discussion about SOA versus ROA is given in the "Status Report on Infrastructures for Biodiversity Research" (Deliverable 5.1.4), Appendix B

105 Ghosh, Sukumar (2007), *Distributed Systems – An Algorithmic Approach*, Chapman & Hall/CRC

be used by third-party applications. The enterprise application servers are also called platform middleware106. The architecture overview of LifeWatch services presented in Figure 7 is an example of a n-tier architecture.

Relevant technologies

Editor's Note: To Be Completed.

8.4.2 Computer clusters

Computer clusters are groups if linked together closely so that they appear to be a single computer. Cluster nodes are often situated on the same subnet or domain with high bandwidth connections. Clusters are designed for jobs that have been called traditionally "supercomputing". The creation of a cluster can have different goals: High-available clusters improve the availability of services by using redundant nodes, load-balancing clusters improve the performance of the system by distributing transparently user requests, and cluster computing is used primarily for computational purposes. Middleware such as PVM (Parallel Virtual Machine)107 or those based on the MPI (Message Passing Interface)108 standard allow to port computer clustering programs to a wide variety of clusters

The design of a computer clusters can differs on how tightly the nodes are connected, going form a tightly coupled architecture to a peer-to-peer approach. In a tightly coupled architecture, distributed machines work closely together to run a shared process in parallel. All participants play equal roles and are therefore they are called peer. A task is subdivided in parts and distributed to the peers, each result is sent back to a controller node that put it together to a final result. This paradigm is appropriated for collaborative work, instant messaging, and file transfer and is often used for solving storage and latency problems. In a peer-to-peer architecture, (or loosely coupled architecture) all responsibilities are divided uniformly among all machines, also called peers, without having a managing machine. Communication between nodes is often done via a queuing mechanism. Peers can serve as clients or servers and

Relevant technologies

Editor's Note: To Be Completed.

8.4.3 Grid computing

Grid computing is based on computer clusters, but more focused on throughput than in running fewer tightly coupled jobs. The main characteristics109 are that the computing resources are not administered centrally, that open standards are used and a nontrivial quality of service is achieved. The computing resources, also called nodes, in a grid are often situated in heterogeneous environments, distributed geographically, and administrated by unrelated organisations.

---

106 Yefim V. Natis, Massimo Pezzini, Kimihiko Iijima, and Raffaella Favata (2008-04-24). "Magic Quadrant for Enterprise Application Servers, 2Q08". Gartner. Available at https://www.salesforce.com/de/assets/pdf/whitepapers/GartnerMagicQuadrantEAS2008.pdf . Accessed October 2009

107 http://www.csm.ornl.gov/pvm/

108 http://www.mcs.anl.gov/research/projects/mpi/

109 Ian T. Foster and Carl Kesselman.(1998) The GRID: Blueprint for a New Computing Infrastructure.Morgan Kaufmann, San Mateo, CA, USA

Grid environments belong to the so- called Shared Nothing Architectures110 due to their independence. The share nothing architecture is based on independent, self-sufficient distributed nodes having nothing in common. Hierarchical systems, the World Wide Web and the Google Search Engine are typical example of it. Google adapted the concept for the horizontal partition of its search engine, calling it sharding.

Each node of a grid system can itself be a parallel system. Workloads on a grid consist of many independent jobs or packets performed on nodes that do not share data between the jobs during the computation process. Each node use to maintain the state of the process performed there.

A grid can be realized as an infrastructure layer, allowing conventional programs to run in multiple machines. This is considered a grid middleware (e.g. Globus Toolkit111, UNICORE112, and gLite113, implemented and promoted by the European leading computer project EGEE114).

Specialized grid infrastructures are:

- Data grids, which deal mainly with the controlled sharing of large amount of distributed data. They are also called distributed data caching.
- Utility computing or computational grids is the offer of resources such as storage and computing power attached to a pay-per-use model.
- Collaboration or scientific grids are used in scientific research and provide a research computer infrastructure, containing high performance computing, visualisation, and collaboration technologies, to a range of scientific communities. The community developed user interfaces (e.g. portals or applications) to access infrastructure resources are often called Scientific Gateways. Examples of global collaboration grid projects are the European EGEE grid infrastructure115, the European Earth Science Grid, D4Science116, and the US-American TeraGrid117, among others.

Relevant technologies

Editor's Note: To Be Completed.

- D4Science
- neuGRID
- myGrid

8.4.4 Cloud computing

Cloud computing is another paradigm based on abstracting technology infrastructure from the user. The term cloud is metaphor for the internet. It involves the provision of dynamically scalable and often virtualized resources as a service over the internet. Major providers of cloud infrastructures are, IBM, Amazon, Google and CloudCamp.

---

110 See Stonebraker, M. (1986) The case for Share Nothing, available at http://db.cs.berkeley.edu/papers/hpts85-nothing.pdf (November 2009)
111 http://www.globus.org/toolkit/
112 http://www.unicore.eu/
113 http://glite.web.cern.ch/glite/
114 http://www.eu-egee.org/. In 2009 the resources coordinated by EGEE will be managed by EGI (http://web.eu-egi.eu/)
115 http://www.eu-egee.org
116 http://www.d4science.eu/
117 http://www.teragrid.org/

There is often a confusion between virtualisation, grid, utility computing , autonomic computer and cloud computing: Virtualisation within the hosting environment is comparable to cloud computing, or, in the opposite way, the majority of existing cloud infrastructures consist of reliable services built on servers with different levels of virtualisation, see Figure 48. Many cloud computing deployments depend on grid technologies, have autonomic characteristics and uses a pay-per-use model like in utility computing.

The main characteristics are:

- Agility and scalability: On demand and inexpensive re-provision of infrastructure resources on a fine-grained, self-service basis near real time
- Device and location independence: users access systems using a browser regardless of their location and devices (PC, mobile, etc)
- Multi-tenancy enables sharing of resources, mainly provided by third-parties, and costs across a large pool of users
- Multiple redundant sites improves reliability

The cloud architecture is based on web services as system interface. Cloud services tend to fit into the category of Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS), or Infrastructure-as-a-Servce (IaaS) that can be encompassed in a stack in analogy to the existing application stack as shown in Figure 48.

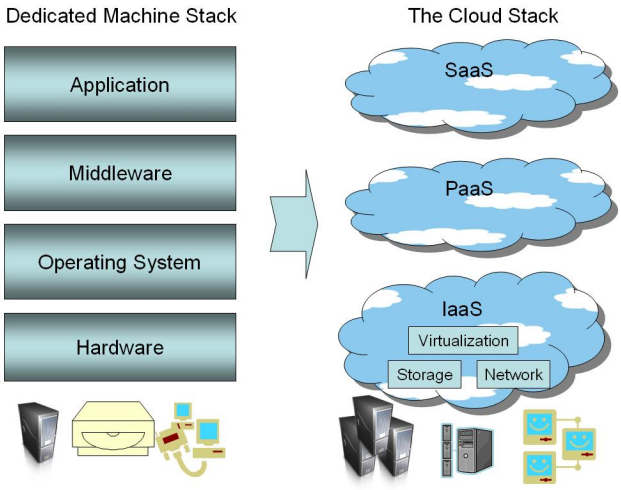


Figure 48 Comparison between the existing application stack and the cloud stack118

Examples of services are:

- IaaS: Computer infrastructure is delivered as a service, typically using a platform virtualisation.
  - IaaS for computers: Amazon EC2, Rackspace
  - IaaS for storage: Amazon S3, MobileMe, GoogleDrive.
  - IaaS for elasticity: Rightscale
  - IaaS for bandwidth: Limelight, Amazon CloudFront
- PaaS: Computing platform is delivered as a service to facilitate the deployment of applications, appearing as an integrated solution over the web. It includes workflow facilities for application

118 Nati Shalom's Blog: What is the cloud? An end-user view, available at http://natishalom.typepad.com/nati_shaloms_blog/2009/02/there-has-been-a-continues-hited-debate-around-the-definition-of-cloud-computingis--it-just-virtualization-is-it-grid-is.html (November 2009)

design, development, testing, deployment, and hosting as well as for team collaboration, database integration, security, etc.
    - Salesforce, Apprenda, Google App Engine, Microsoft Azure
- SaaS: Software provider licences applications to customers to be used as a service on demand.
    - Google Apps, Zoho, Acrobat.com, iWork.com, Kasflow, FreeAgent, WorldPress.com, MobileMe, Salesforce.com

Relevant technologies

Editor's Note: To Be Completed.

8.4.5 Space-based architecture

Space-based architecture (SBA) first goal is to achieve linear scalability and high availability of stateful, high performance applications. This software architecture pattern follows many of the principles of REST, SOA and Event-Driven-Architecture (EDA).

Space-based applications are built from a set of self-sufficient independent units, known as processing-units (PU). A PU is a combination of data, processing services, and messaging support. This architecture creates the illusion of one single address-space. Resources in a PU are replicated transparently on demand into other processing units (using virtual machines) achieving linear scalability.

Space Based Computing is based on four actions:

- Write: Writes a data object to the space
- Notify: generates an event on data updates
- Read: reads a copy of a data object
- Take: reads a data object and deletes from the space

The combination of the actions can be interpreted as data caching or data grid (write + read), messaging (write + notify), and parallel processing (write + take) and are executed using a virtual middleware. The virtual middleware is a common runtime and clustering model used across the entire middleware stack, typically composed of a messaging grid for the transaction and communication flow, a data grid for distributing memory spaces, and a processing grid for parallel processing of events among different services.

An approach based on Service-Level-Agreements (SLA) definitions and policies allows the deployment of applications in a dynamic pool of machines.

Relevant technologies

Editor's Note: To Be Completed.

8.4.6 Autonomic computing

Autonomic computing is a system design for self-managed systems covering the following areas119.

- Self-Awareness: an autonomic system knows itself and is aware of its state and its behaviours.

---

119 http://www.caip.rutgers.edu/TASSL/Papers/ac-paradigm-jcc-06.pdf

- Self-Protecting: an autonomic system is equally prone to attacks and hence it should be capable of detecting and protecting its resources from both internal and external attack and maintaining overall system security and integrity.
- Self-Optimizing: an autonomic system should be able to detect performance degradation in system behaviours and intelligently perform self-optimization functions.
- Self-Healing: an autonomic system must be aware of potential problems and should have the ability to reconfigure itself to continue to function smoothly.
- Self-Configuring: an autonomic system must have the ability to dynamically adjust its resources based on its state and the state of its execution environment.
- Contextually Aware: an autonomic system must be aware of its execution environment and be able to react to changes in the environment.
- Open: an autonomic system must be portable across multiple hardware and software architectures, and consequently it must be built on standard and open protocols and interfaces.
- Anticipatory: an autonomic system must be able to anticipate, to the extent that it can, its needs and behaviours and those of its context, and be able to manage itself proactively to be considered when specifying a service platform

Relevant technologies

Editor's Note: To Be Completed.

8.5 Technology options concerning particular capabilities

This section addresses particular technologies that are somewhat platform independent. Each subsection starts with an outline of requirements partly derived from the engineering policies followed by recommendations for technologies that satisfy – at least partly - the requirements.

8.5.1 User management, authentication, authorisation, and accounting

At present stage, Shibboleth[120] seems to be the technology that best matches the policies expressed in the engineering viewpoint. A shortcoming is that Shibboleth is an institution-centric in that organizations must act as trusted verifier of the authentication data. This may present problems for users working outside any institution, which may demand for a more user-centric identity solution.

OpenID[121] provides a user-centric approach putting users in control of how their identity is managed and used online. In contrast to Shibboleth where an institution, typically the home institution, asserts that a you are who you say you are, for OpenID a content provider has to trust users to be who they say they are.

LifeWatch may combine both these technologies to provide a staged access to resources: more critical resources may only be accessed only by a more institution-centric approach such as Shibboleth while OpenID may be sufficient for other resources.

[120] http://shibboleth.internet2.edu/
[121] http://openid.net/

8.5.2 Workflows

A recommendation of a particular technology for handling workflows is difficult due to a lack of standards for workflow languages. The most widely accepted standard is BPEL (Business Process Execution Language) that, however, suffers from a number of shortcomings concerning scientific computation[122]:

- BPEL is not in tune with the kind of data-flow, time-based computation models often used for manipulation of data sets and simulation.
- It does not support stepwise interaction scientists or accessing different types of computational resources other than web services.
- It does not match well with the users skills in that its usage requires more expert users than researchers in biodiversity may be expected to be.
- There is no support of provenance.

Though changing or extending parts of the BPEL specification or by customized environments may mitigate these issues, the present situation is not such that a BPEL based workflow environment can be recommended.

Criteria next to standards are existence of a user basis and an active developer community. On this basis, Kepler[123] and Taverna[124] seem to be the technologies of choice. Triana[125] and VisTrails[126] are technologies built on the same principles but seem to lack both at present. All these systems represent workflows by flow graphs based on a data flow paradigm. These systems are compared below on basis of the major requirements derived from the engineering viewpoint. Other technological options that may emerge should be compared on basis of the same criteria. The comparison is done in terms of a table with requirements being

General

a) Separation of authoring and enactment of workflows
b) Authoring as Web application
c) Integration of external tools
d) Service discovery
e) Support for Grid or Cloud
f) Capabilities to control and monitor the execution of the workflow
g) Reusability of workflows as services
h) Provision and maintenance of the metadata descriptions of services and workflows
i) Generation of provenance information
j) A modular design: plug-in architecture and support for toolboxes
k) Support of the access policies and security constraints
l) Support for workflow repository
m) Support for data access protocols
n) Support for versioning

User interface

o) Support of hierarchy: workflows as components
p) Composition and linking based on the descriptions of inputs and outputs
q) Feedback about (semantic) compatibility of inputs and outputs

---

[122] A. Goderis, P. Li, C. Goble. Workflow Discovery: Requirements from E-Science and a Graph-Bases Solution. International Journal of Web Services Research, Vol. 5, Issue 4, 2008

[123] http://kepler-project.org/

[124] http://www.taverna.org.uk/

[125] http://www.trianacode.org/

[126] http://www.vistrails.org/

r) Support for user interaction during enactment

	Kepler	Taverna	Triana	VisTrail
a	?	✓	?	?
b	✘	✘	✘	✘
c	✓	✓	✓	
d	✓	✓	?	✓
e	✓	?	✓	✘
f	?	✓	✓	?
g	?	?	✓	✘
h	?	?	?	?
i	✓	✓	?	✓
j	✓	✓	✓	?
k	?	?	?	?
l	✓	✓	?	✓
m	✓	✓	✓	?
n				✓
o	✓	✓	✓	
p	✓	✓	✓	?
q	?	?	✓	✘
r		✓	✓	?

The lack of a Web interface is notable in all cases. LifeWatch may develop such an interface for representing/editing workflows. This may offer the chance to find a technology agnostic frontend that translates to the languages accepted by the different enactment machines (BPEL, Scufl, MomL).

8.5.3 Semantics

Many of the technologies related to semantic capabilities are under development though some standardisation activities are under way. Hence, technology recommendations are restricted to standardisations. An overview of the technologies and their status can be found in the LifeWatch Status Report, Section 8.

Ontology

Though not normative, LifeWatch recommends the use of RDF and OWL-DL for ontologies. The most widely used tool to edit and visualize ontologies is Protegé.[127]

Basic structures content of concept schemes such as thesauri, classification schemes, subject heading lists, taxonomies, folksonomies, and other similar types of controlled vocabulary may be expressed using modelling of Simple Knowledge Organization System[128] (SKOS)

For modelling web services, the Web Service Modeling Ontology (WSMO) may be recommended.

[127] Commercial support is available (e.g. by Ontoprise).
[128] http://www.w3.org/TR/skos-primer/

SKOS—Simple Knowledge Organization System—provides a model for expressing the basic structure and content of concept schemes such as thesauri, classification schemes, subject heading lists, taxonomies, folksonomies, and other similar types of controlled vocabulary.

Annotation

Annotation depends on the basic language support. In case that documents are XML-based, SAWSDL is recommended as annotation standard. SAWSDL supports the annotation of XLM schemata as well as of web services defined in terms of WSDL.

In case of RESTful web services a micro format such as MicroWSMO might be used, or a version WADL enhanced by the annotation scheme promoted by SAWSDL.

Discovery

8.5.4 Provenance

Technologies concerning provenance are either bound to particular systems, e.g. workflows, or are in an experimental status. The Open Provenance Model is a possible future standard that LifeWatch shall support in that the systems chosen should support the Open Provenance Model or provide a translation of an internal presentation of provenance to the Open Provenance Model. LifeWatch shall provide basic provenance information as meta-information model. The extent of this information is to be determined according to needs.

8.5.5 User interface

The technologies for user interfaces of web-based applications are in flux. The HTML5 standard[129] dramatically increase the capabilities of the web. In particular the canvas element enhanced by object models as those defined by webGL[130] for 3D graphics will open the scope for totally new web applications up to gaming and virtual realities without demanding for particular plug-ins. The advent of Google Chrome OS[131] may mark the starting point of the dawn of desktop computing but future developments are unpredictable.

The aim of LifeWatch implicit in the choice of the SOA paradigm is independency from a particular presentation layer though this may be wishful thinking if it comes to sophisticated user interfaces as, for instance, the handling of workflows demand for. From the present point of view however, it seems to be likely that there will be a trend to web-based user interfaces. The advantages are obvious: update and versioning problems can be avoided though they creep in due browser and plug-in versioning.

There a number of technologies that support the creation of user interfaces as rich web applications (as indicated in Section 11.12 of the accompanying status report) but, since many of the technologies are brand-new, their future is incalculable. A multiplatform language like haXe[132] might help but typically the necessary libraries are platform dependant. The development here is not as far as for desktops where cross-platform libraries allow writing applications that 8almost) have the touch and feel of the respecitive desktop environments.

---

[129] http://www.w3.org/TR/html5/
[130] http://www.khronos.org/webgl/
[131] http://en.wikipedia.org/wiki/Google_Chrome_OS
[132] http://haxe.org

As a consequence, no recommendation can be given of what technology should be used for the User interface layer.

Blank Page

Appendix A. Summary of the ORCHESTRA Reference Model

A.1 Introduction

Adherence to standards is one issue, best practice to interpret standards a second. In order to minimise effort LifeWatch builds on approved best practices as provided by the community.

The ORCHESTRA architecture (http://www.eu-orchestra.org/) is an architectural framework committed to standards set by ISO/IEC 10746 and ISO/DIS 19119. The Reference Model for the ORCHESTRA Architecture (RM-OA) achieved best practice status within the OGC consortium (http://www.opengeospatial.org/standards/bp). It is an extension of the OGC Reference Model and contains a generic specification framework for the design of geospatial service-oriented architectures and service networks. The ORCHESTRA framework has been successfully used in European projects such as:

- SANY Sensors Anywhere (IST FP6 Integrated Project, http://www.sany-ip.eu/): The project focuses on interoperability of in-situ sensors and sensor networks and will provide a service-oriented architecture for environmental sensor networks. Applications are in the area of risk analysis of, for instance, air and water pollution.
- The GEOSS, INSPIRE and GMES an Action in Support (GIGAS, http://thegigasforum.eu/) promotes the coherent and interoperable development of the GMES, INSPIRE and GEOSS initiatives through their concerted adoption of standards, protocols, and open architectures.

Because of the maturity of the ORCHESTRA design, its conformance with international standards as well as its adaption by major European and international initiatives the strategy of LifeWatch is to exploit the ORCHESTRA architecture and to build the LifeWatch architecture on top of it, extending the ORCHESTRA Reference Model and adapting it when appropriate. To be reasonably self-contained, an overview of the Reference Model for the ORCHESTRA Architecture (RM-OA) is given subsequently.

A.2 The ORCHESTRA Approach

The challenge of LifeWatch is to build an IT-infrastructure that provides seamless access to resources (information and services) across organisational and technical borders. There are several fundamental challenges that may be classified into the four following categories:

- Syntactic Interoperability: how to overcome technical heterogeneity of all sorts.
- Semantic Interoperability: how to overcome ambiguities and different interpretations
- Organisational Context: how to overcome organisational cross-border situations of all kinds
- Generics: how to deliver sustainable and reusable and independent concepts and components

The ORCHESTRA approach, although focussed on risk management as its application area, provides a generic framework for addressing these challenges.

The ORCHESTRA process model promotes an incremental, iterative approach for the analysis and design phases. It distinguishes between an abstract service platform specified independently of any middleware technology and a concrete service platform that is implemented on a specific middleware (see Figure 2)

- The abstract design phase leads to platform-neutral specifications following the rules of the abstract service platform provided by the ORCHESTRA Reference Model. They represent the functional requirements (abstract service specification), informational requirements (information model), and non-functional requirements (specification of the quality of service (QoS)) of the problem domain.

- The concrete design phase maps the abstract specifications to a chosen concrete service platform. In the current ORCHESTRA project, this is the ORCHESTRA Web Services platform consisting of the rules of the W3C Web services and a profile of the Geography Mark-up Language (GML) as the current (mainstream) service platform technologies for geospatial applications.
- In the engineering phase, the platform-specific components are organised into service networks taking into account the QoS requirements and translating them into operational policies.

Since these design phases are often interlinked, a typical design follows a middle-out strategy: services are sometimes defined relative to a specific platform and only then abstracted.

The ORCHESTRA Architecture (OA) provides a generic modelling toolbox in terms of predefined but generic information and service types (OA services) upon which the functional and informational user requirements may be mapped.

An ORCHESTRA Application Architecture is an instantiation of the ORCHESTRA Architecture by inclusion of those thematic aspects relevant to a specific application area. The concepts for such an application stem from a particular application domain. The ORCHESTRA Reference Model provides a conceptual model with detailed rules about how to specify an information model and a service model that fit to the predefined ones and adapt them to so-called thematic models and services for a particular application area. The relationship between some ORCHESTRA Application Architecture and the ORCHESTRA Architecture is shown in Figure 49

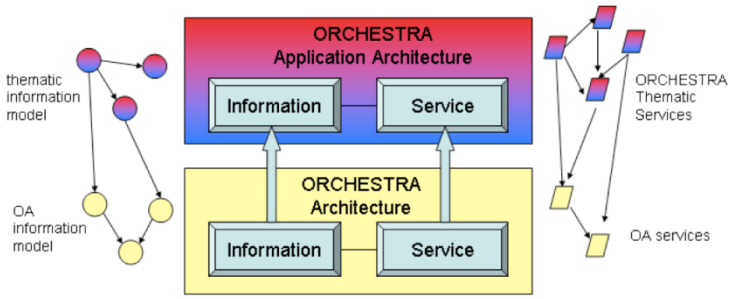


Figure 49: ORCHESTRA Application Architecture (Source: RM-OA)

The ORCHESTRA Implementation Architecture for the ORCHESTRA Web Services platform complements the abstract architecture. Here, ORCHESTRA delivers a software toolbox comprising implementation specifications and implementation components derived from and compliant with the abstract specifications. For the thematic information and service models of an application architecture, tools are provided to map them to this platform.

General elements of the ORCHESTRA architecture are (cf. Usländer, Denzer, Güttler: „Open Service-oriented Architecture for Environmental Risk Management Applications“133 ISESS 2007 Symposium134 (Prague, 22-25 May 2007):

- A process model compliant with an ISO standard (RM-ODP) and tailored to the design and engineering of geospatial SOAs.
- A reference model for the design of geospatial SOAs.
- An open abstract architecture containing design rules for information and service models.
- Specifications, on both the abstract level and specific to the W3C Web services platform, of the most important generic architecture services (derived from but not restricted to the needs of environmental risk management application.

133 http://www.eu-orchestra.org/docs/20070522-OrchestraPaper-ISESS2007-ORCHESTRA_Architecture.pdf
134 http://www.isess.org/

- Software engineering components, mostly offered under an open source license, for the development of service networks including
- A Java-based software framework called OSCF (ORCHESTRA Service Container Framework) that comprises APIs for common service functionality (e.g. the service capabilities, access control) and therefore facilitates the implementation of additional services according to the ORCHESTRA approach,
- Implementation of indispensable architecture services integrated into the OSCF
- Adapters to industry standard services (e.g. the OGC Catalogue service with both the ebRIM and the ISO Application Profile),
- Design support for the mapping of service and information models from the abstract level (UML) to the Web services platform (WSDL, XML/GML),
- A Java-based software framework for the integration of source systems (e.g. for relational databases) into an ORCHESTRA service network, and
- Utility applications (e.g. for user management, service monitoring, catalogue navigation)

Figure 1 in Section 3 indicates the relationships of the ORCHESTRA architecture to the various standards concerned

A.3 ORCHESTRA Service Network

The operational components of an ORCHESTRA Service Network are the ORCHESTRA Service Instances (OSIs). OSIs offer their functionality and interact among each other according to the ORCHESTRA protocol, i.e. the set of the ORCHESTRA rules given by the ORCHESTRA Meta-model (OMM) as described below. OSIs are organised in the following functional domains as in Figure 50 below

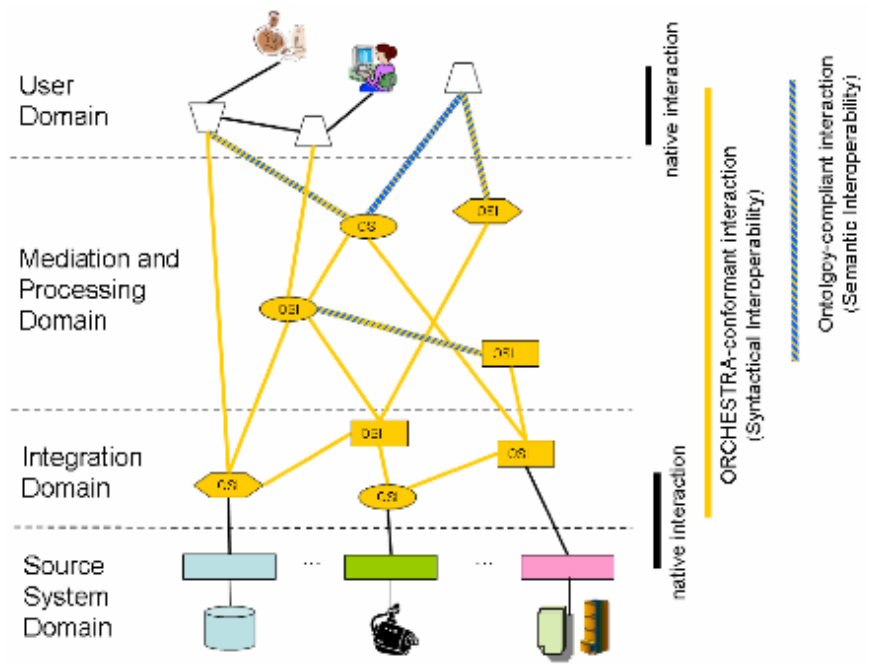


Figure 50: Functional Domains of an ORCHESTRA Service Network (Source: RM-OA)

- Software components in the User Domain provide the interface to a user component (a human or another software component). When interacting with an OSI, they have to use the ORCHESTRA protocol.
- OSIs in the Mediation and Processing Domain provide the main functional part of an OSN. They mediate the service calls from the User to the Integration Domain based on meta-information ex-

changed with the components of the Integration Domain (e.g. by means of a publishing or a retrieval pattern).

- OSIs in the Integration Domain provide support for the integration of source systems into an OSN. The OSIs in this domain have two-sided interfaces. On the one hand, they interact with other OSIs according to the ORCHESTRA protocol. On the other hand, they interact with the components of the Source System Domain according to their native protocol.
- The Source System Domain incorporates the systems and system components of an application area to be integrated into an OSN. In practice, this means that their data and functionality have to be wrapped with an ORCHESTRA-compliant service interface.

A.4 Abstract Service Platform

On the level of the abstract service platform, the ORCHESTRA Architecture provides the following elements:

- A description framework and document templates for the textual specification of interface and service types.
- A coherent set of rules to specify interface, service and feature types in UML and to organise them in service and information models. This rule set is referred to as ORCHESTRA Meta-model (OMM).
- A specification of important feature types that may be re-used and refined in information models.
- Specifications of a series of generic interface and service types that may (and should) be re-used by service modellers in the design of an application area.

Supported service types are listed in Table 2below.

Table 2: Supported service types of ORCHESTRA

Service Type Name	Description
Basic Interfaces	Interface types enabling a common architectural approach for all ORCHESTRA Services: - Self-description of service instances (capabilities) - Synchronous and asynchronous interactions - Transactional support Furthermore: predefined exception types
Authentication Service	Proves the genuineness of principals (i.e. the identity of a subject which may be a user or a software component) using a set of given credentials. Selected authentication mechanism is up to implementation specification.
Authorisation Service	Provides an authorisation decision for a given authorisation context. Selected authorisation paradigm is up to implementation specification.
Catalogue Service	Ability to publish, query and retrieve descriptive information (meta-information) for resources (i.e. data and services) of any type, not tied to a particular schema of a meta-information standard (e.g. ISO 19115) A Catalogue Service supports application schemas for meta-information that are designed according to the ORCHESTRA rules May be used as a data catalogue, service registry, or both. May be based on other standard catalogues like OGC Catalogue or UDDI.
Document Access Service	Supports access to documents of any type (textual documents, images,). A document is considered to be a specific kind of a feature type.

Map and Diagram Service	Visualizes, symbolizes and enables geographic clients to interactively visualise geographic and statistical data. Transforms geographic data (vector or raster) and/or numerical tabular data into a graphical representation using symbolization rules. The main output of this service is an image document that may be a map, a diagram or a thematic map (visualization of the spatial distribution of one or more statistical data themes).
Feature Access Service	Selection, creation, update, and deletion of feature instances and feature types available in a service network. Features provided are instances of a certain feature type defined in an ORCHESTRA Application Schema. Interface may be re-used by more specific access services using interface inheritance.
Name Service	Encapsulates the implemented naming policy for service instances in a service network, e.g. creates globally unique service instance names using a defined naming policy. Important if several service networks across different platforms are to be interconnected.
Sensor Access Service	Basic interface for accessing sensor data, configuring a sensor and publishing sensor data.
Service Monitoring Service	Provides an overview about service instances currently registered within service network, e.g. 1. Actual status (e.g. running, stopped, offline) 2. Statistical information (e.g. average availability, response times)
User Management Service	Creates and maintains subjects (users or software components) including groups (of Principals) as a special kind of subjects.

A.5 Concrete Service Platform

The ORCHESTRA Meta-model provides rules describing how to map the abstract specifications to a concrete service platform. There are software tools available from ORCHESTRA that support this mapping process for the ORCHESTRA Web services platform. In this mapping process, UML information models have to be translated to XML/GML whereas UML interface and service models have to be mapped to WSDL documents. For the service types listed above, ORCHESTRA also provides corresponding implementation specifications and implementations, most of them integrated in the common ORCHESTRA Service Container Framework.

A.6 The ORCHESTRA Reference Model (OA-RM Section 5.3)

Figure 51 relates the Reference Model for Open Distributed Processing (ISO/IEC 10746) to the ORCHESTRA architectural design process.

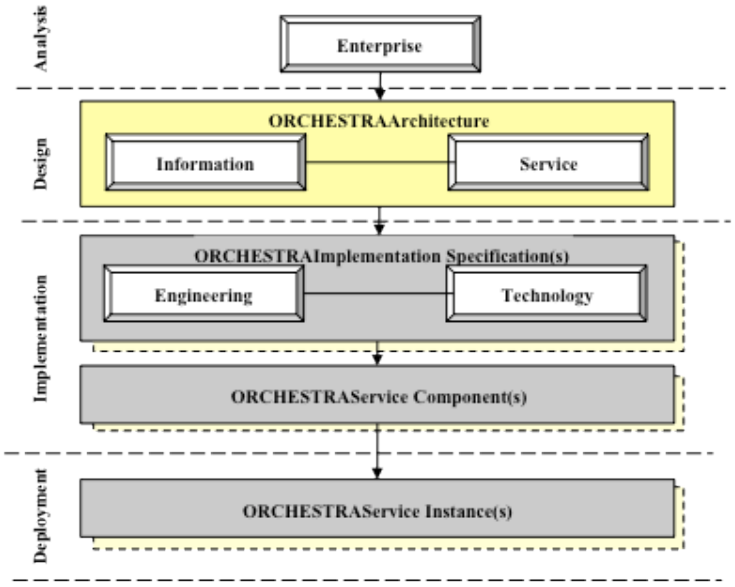


Figure 51: The ORCHESTRA Reference Model (Source: RM-OA)

The Analysis Phase leads to the specification of the Enterprise Viewpoint that defines goals, scope, and policies. The Design Phase leads to the specification of the overall Architecture, and the Implementation Phase to the Implementation Specifications that are implemented by Service Components.

The ORCHESTRA Architecture is a platform-neutral specification according to the requirements of ISO 19119:2005. Platform-neutral means that the architecture is independent of particular features and properties such as web or grid services. Platform-neutral specifications use the conceptual modelling language UML. The advantage of a platform-neutral specification is that it provides a reference with regard to different platforms and thus facilitates interoperability between different platforms. The Architecture provides a consistent view of the Information Viewpoint and the Computational Viewpoint as promoted by the Reference Model for Open Distributed Processing (ISO/IEC 10746).

In a nutshell, the Information Viewpoint specifies the modelling approach for all categories of information including meta-information and the Computational or Service Viewpoint (in the terminology of ORCHESTRA) provides the specification framework for as well as platform-neutral specifications of generic services.

The Implementation Phase leads to the platform-specific Implementation Specifications implemented as LifeWatch Service Components. The information models and services of the Architecture (in UML) are mapped to a schema language (e.g. XML/GML) that fits to the selected platform. ORCHESTRA's Technology and Engineering Viewpoints provide guidelines, requirements, and rules for mapping to a platform and for designing a Service Network, i.e. a communication network that connects deployed instances of services.

A.7 Integration of Source Systems

The ORCHESTRA Architecture provides mechanisms to integrate "legacy" or "source systems" (the terminology of ORCHESTRA). Source system integration is defined as the process of transforming External source systems into ORCHESTRA Source Systems, i.e. source systems that provide their data and functions through an ORCHESTRA-conformant interface. The RM_OA specifies rules for the transformation.

A.8 Interoperability Between Different Service Platforms

ORCHESTRA does not yet support interoperability between service networks operating on different platforms. Interactions between service instances that belong to different platform domains should be made possible by the provision of service platform gateways. An example for such a situation is a gateway that maps between a CORBA-based platform and Web Services.

Blank page

Appendix B. The LifeWatch meta-model approach

B.1 Introduction

To ensure syntactically and semantically interoperability between information-models defined for or used within the LifeWatch infrastructure the Reference Model provides a set of elementary concepts based principally on standards and a set of rules to be applied when defining information models. The rules and the elementary concepts define the LifeWatch meta-model that is firstly a mechanism aiding the representation of real world concepts for three main goals:

- The information meta-model supports the modelling of data concepts (mainly expressed as features and their properties and relationships) as application schemas.
- The service meta-model provides the elements for defining and describing services
- The meta-model for information is based on the concepts and rules of the information meta-model for the development of purpose-oriented meta-information models, but extends these by outlining the LifeWatch purposes and by providing a set of standards-based elements and specific rules to be used when specifying meta-information.

This introductory section gives an overview of the basic principles of the meta-model approach, namely the meta-model concepts of the ORCHESTRA-RM and the INSPIRE Implementing Rules.

B.1.1 Basic rules derived from the ORCHESTRA Meta-Model

The ORCHESTRA meta-model is used to support the generic architecture specified on the ORCHESTRA-Reference Model, which does not prescribe specific information models nor configurations of service instances. The ORCHESTRA meta-model (OMM) consists in two parts, (1) the OMM-Information to be used for the specification of application schemas for information models and meta-information models, and (2) the OMM-Service with rules about how to specify service interfaces and a set of proposed architectural services. Both parts should be interrelated ensuring that:

- The structure of input and output parameters of service interface operations has to follow the rules of the OMM-Information and
- The built-in operations on feature types have to obey the rules of the OMM-Service.

General OMM rules, that are also followed by the LifeWatch meta-model approach can be summarised by:

- The OMM-Information is an extension of the General Feature Model defined in ISO 19109, thus it is a meta-model for feature types, and mandates the usage of UML 2.0 as conceptual schema language.
- The OMM-Information provides rules for the usage of value domains for the attributes of feature types (attribute types)
- The OMM-Information dies not restrict the use of ISO 19115 for attribute types representing meta-information as specified by ISO 19109, but proposes a purpose-oriented definition of meta-information attributes and value domains
- The OMM-Service provides a platform-neutral approach to specify Service Types as a collection of Interface Types, which comprises a set of operations as interaction unit between a consumer and a service instance. It does not provide a formal specification of Service Types. The LifeWatch services meta-model

B.1.2 INSPIRE implementation rules

The INSPIRE Directive135 refers to detailed common Implementing Rules (IR) on a number of topics to ensure that the spatial data infrastructures of member states are compatible and usable in a community an trans-boundary context. The European Commission formalized Implementing Rules commission decision or regulation legally binding for a coherent implementation. This section lists current areas relevant to LifeWatch, for which the Directive specifies common implementing rules and sketches the approach to achieve compliance to the regulations136 through the LifeWatch infrastructure.

Metadata

Regulation	**Guidance Documents**
INSPIRE Metadata Regulation (12.04.2008)137	INPIRE Metadata IR: Technical Guidelines based on EN ISO 19115 and EN 19119
LifeWatch compliance	
• LifeWatch should contribute to biodiversity themes (Annex III) by contributing to the meta-data model specification process and providing access services to those meta data • LifeWatch should include the terms specified in Part A of the Annex138 in the meta-information model for discovery and/or in the core ontology and refer to them whenever possible • LifeWatch should provide a wrapper service providing meta-information conformant to the INSPIRE ISO profiles	

Data Specifications

Regulation	**Guidance Documents**
Not available	Provide guidelines for the specification of spatial data themes for the Annex I-III, (available for Annex I). The guidelines supplement implementation rules for inter-operability of spatial data sets and services and provide support for the implementation.
Relevant themes for LifeWatch	
Annex I: • Protected Sites	Annex III: • Bio-geographical regions • Habitats and biotopes • Species distribution
LifeWatch compliance	
LifeWatch should provide services to access biodiversity data as specified by the inspire themes: • Wrappers should offer access to the data according to the defined application schemas (at least the simple application schemas) for the relevant themes • The feature types, attributes, and relationships for the relevant themes should be accessible via a fea-ture catalogue • Spatial and temporal descriptors should contain elements defined according to the application sche-mas for the themes in Annex I: (geographical names, geographical grid systems, geographical refer-	

135 See http://inspire.jrc.ec.europa.eu/index.cfm

136 The regulations reflect the status at the time of writing the present document and were available at http://inspire.jrc.ec.europa.eu/index.cfm/pageid/47 (October 2009)

137 http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:32008R1205:EN:NOT

138 The terms defined are: “character string”, “free text”, “lineage”, “metadata element

ence systems, and addresses)
- Services can be provided for the generation of European spatial data reports and Natura2000 reports
- Visualisation services should offer interactive mapping functionalities for data conform to the specified application schemas, offering the specified layer information.
- Data quality elements used in the spatial data themes should be included in the meta-information models

Network Services

Regulation	Guidance Documents
Draft Regulation on INSPIRE Discovery and View Services. 02.06.2009139	Draft Technical Guidance for • Download Services • Coordinate Transformation Services • View Services • Discovery Services
LifeWatch compliance	
• LifeWatch should provide wrapper interface to meet the specification of INSPIRE Network Services for Discovery, View, Download and Transformation in order to accessed LifeWatch data conform to the biodiversity related INSPIRE themes through the INSPIRE GeoPortal and in order to access services available at the INPIRE GeoPortal through the LifeWatch infrastructure. • The concepts defined on the regulation should be included in the ontology in order to comply with the INSPIRE vocabulary meta-information elements and capabilities. The regulation defines the following terms: o Initial Operating Capabilities o Performance o Capacity o Availability o Response time o Service request o INSPIRE Metadata Element o Publish o Natural Language o Collect o Layer • The LifeWatch services complying the INSPIRE Network Services regulation shall conform the requirements for quality of services regarding performance, capacity and availability • The Discovery and View Service should conform the interfaces (exchanged elements and operations) defined on Annex II and III respectively of the regulation.	

Data and Service Sharing

Regulation	Guidance Documents
Draft Regulation on INSPIRE Data and Service Sharing. 05.06.2009140	Guidelines and good practice documents
LifeWatch compliance	

[139] http://ec.europa.eu/transparency/regcomitology/searchform/DocumentDetail.cfm?vHms+N/xSr/4vbbVbuLqZLtyid2hPzM6zGr14hlwr48gQQ3qxtO4vnSJh7flF4ct

[140] http://ec.europa.eu/transparency/regcomitology/searchform/DocumentDetail.cfm?o1zZRSuBXDWNJfglInPW3K6Mfjz5PjGANUUGptseSCs/zBk6+Yr2dztb8GfTzcg1

- LifeWatch access services for the relevant INSPIRE themes should provide the access conditions (e.g. in form of licences and terms of use) as well as information for evaluation and use, on the mechanisms for collecting, processing, and producing data (provenance information), and quality control.
- INSPIRE intends to publish documents for regulation of the topic criteria (transparency, framework agreements, coordination, charging mechanisms, public access, etc.) by the end of 2009.

B.2 The information meta-model

B.2.1 The ORCHESTRA Meta-Model for Information as basis

This section sketches the relevant components of the ORCHESTRA Meta-Model for Information for creating LifeWatch information models, adding LifeWatch specific components when possible and predictable. The presentation follows that of ORCHESTRA in format and content. In general, the description differentiates between ORCHESTRA Basic Data Types that have a standardised definition by a standardisation body (e.g. ISO 191xx series), ORCHESTRA predefined (meta) types, LifeWatch predefined (meta) types, and user-defined types. ORCHESTRA predefined (meta) types are referred to as OMM Types and are prefixed with OMM_. LifeWatch predefined (meta) types are referred to as LW Types and are prefixed with LW_.

B.2.2 Basic data types

Basic data types are those defined outside of ORCHESTRA documents, e.g. in the ISO 191xx series. Table 3 lists the basic data types with a reference to the origin standardisation document.

Table 3: Basic data types

Type Names	Origin	Brief Description
Real	ISO 19103 §6.5.2.5	A signed real (floating point) number consisting of a mantissa and an exponent. (not necessarily the exact value as the common implementation of a Real type uses base 2)
Integer	ISO 19103 §6.5.2.3	A signed integer number, Exact with no fractional part.
Decimal	ISO 19103 §6.5.2.4	A number type that represents an exact value as a finite representation of a decimal number. (Unlike real, it can represent 1/10 without error)
Binary	ISO 19118 §A.5.2.1.14	Finite sequence of arbitrary binary data
Any	ISO 19103	The root of all classes. Often not an actual class implementation, it essentially is used where the target class of a member name is not known.
CharacterString	ISO 19103 §6.5.2.7	Type representing a simple string. The whole string has a single specific encoding, retrievable from the string.
CountryCode	As will be defined in ISO/TS 19139	List of country identifiers
LanguageCode	As will be defined in ISO/TS 19139	List of language identifiers
CharacterSetCode	ISO 19103 §6.5.2.7	List of character encodings
MD_Character SetCode	As defined in ISO 19115	List of character encodings
PT_Locale	As will be defined	Type combining language, country and encoding

Type Names	Origin	Brief Description
	in ISO/TS 19139	
Localised CharacterString	As will be defined in ISO/TS 19139	A character string with the addition of a field specifying the language of the string
Enumeration	ISO 19103 §6.5.4.2	Defined and closed list of valid mnemonic identifiers
CodeList	ISO 19103 §6.5.4.3	An open Enumeration
Boolean	ISO 19103 §6.5.2.11	A value specifying TRUE or FALSE
Date	ISO 19103 §6.5.2.8	Type representing a date
Time	ISO 19103 §6.5.2.9	Type representing a point in time
DateTime	ISO 19103 §6.5.2.10	Type combining date and time
Set	ISO 19103 §6.5.3.2	Unordered finite collection of non-duplicate objects
Bag	ISO 19103 §6.5.3.3	Unordered finite collection of possibly duplicate objects
Sequence	ISO 19103 §6.5.3.4	Ordered ‘bag-like’ structure

B.2.3 ORCHESTRA Meta-Model Types

The Meta-Model describes the basic concepts (meta-classes) needed for creating information models. Those concepts allow generating information models from data sets by extracting feature types (concepts), having property types (attributes, operations and associations), constraints, and relationships with other concepts (e.g. inheritance or other association). Figure 52 shows the class diagram for the meta-classes, as defined by the ORCHESTRA Meta-Model (source: RM-OA V2, chapter 8.7). The meaning of the meta-classes is the same as the meaning of the meta-classes prefixed by GF_ in ISO 19109 GFM. The architectural principle of "self-describing components" of the RM_OA equally applies to LifeWatch, i.e. the feature type specification for a given feature instance can be obtained through the capabilities of the service instance.

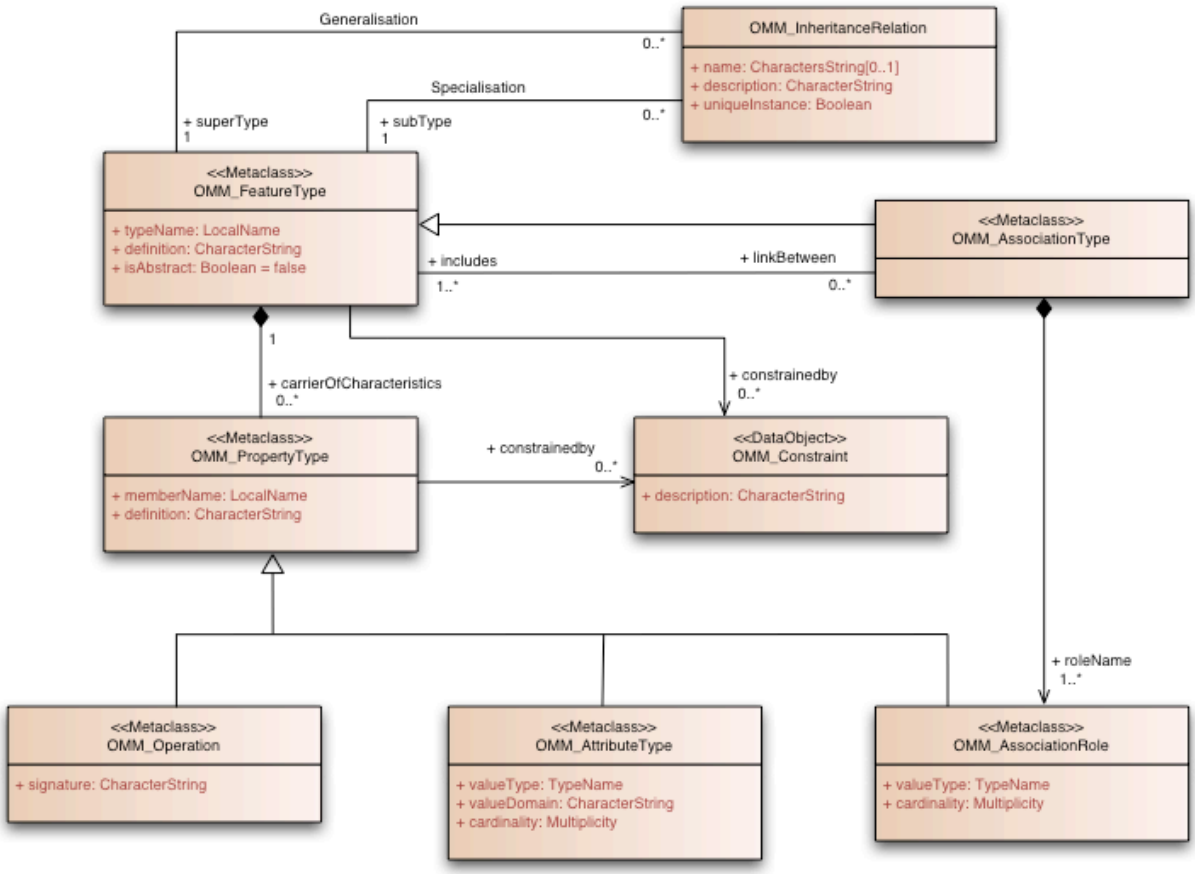


Figure 52: OMM Types (RM_OA V2, Chaper 8.7)

Note: The typing of various attributes in terms of "CharacterString" is a very mild restriction. An example is the description property of OMM_Constraints. ISO 19109 uses this vague type to allow the expression of constraints in a language specific to the implementation environment of applications. The final LifeWatch Meta-Model for types may decide to enhance the typing in that specific language types are used, e.g. RDFGraph.

Similarly, additional features may be nedded. For instance, a feature "formalDescription" may be added at several places being of type "RDF" while the feature "decription" remains of type "Characterstring" to hold plain textual decriptions. The adaptation of metamodels is a LifeWatch policy (see Section 7.6).

The meta-class OMM_AttributeType represents attributes of features, of other attributes or of operations. They carry all static information about a feature. In the ISO 19100 series of standards, some attribute types are of particular interest, namely those carrying temporal, spatial, location and metadata information. The OMM uses the following categories of OMM_Attribute_Types based on the ISO 19100 series as specified in Figure 53:

- Spatial Geometry (ISO19107::GM_Object)
- Spatial Topology (ISO19107::TP_Object)
- Temporal Object (ISO19108::TM_Object)
- Geographic Identifier (ISO19112::SI_LocationInstance)
- Data Quality Information (ISO19115::DQ_Element)
- Metadata (ISO19115::MD_Metadata)

LifeWatch's application schemas should use, whenever it is applicable, those standards. An extension of the above-mentioned types seems obviously when considering taxonomic information. Taxonomy infor-

mation can be defined as a standardized attribute type through its defined characteristics and is a first candidate for a LW_ThematicAttributeType.

The OMM specifies meta-information attributes (OMM_MetaInfoAttributeTypes) as a specialisation of the meta-class OMM_ThematicAttribureType, in contrast to ISO 19109. The reason is that the OMM considers meta-information models depending on a particular purpose and not globally for a feature. Therefore, they are rather attached to a thematic category, than situated on a high level category as in ISO 19109. LifeWatch may extend the OMM_MetaInfoAttributeType in order to add properties to those specified through ISO 19115 according to the LifeWatch purpose-oriented meta-information models.

Figure 53 shows a UML schema for the meta-class OMM_AttributeType. The figure deviates from the original diagram in RM_OA in that LifeWatch may decide to add particular thematic attribute types or meta-information attribute types, depicted green shaded on the right side of the diagram. The need for such extensions should be considered carefully.

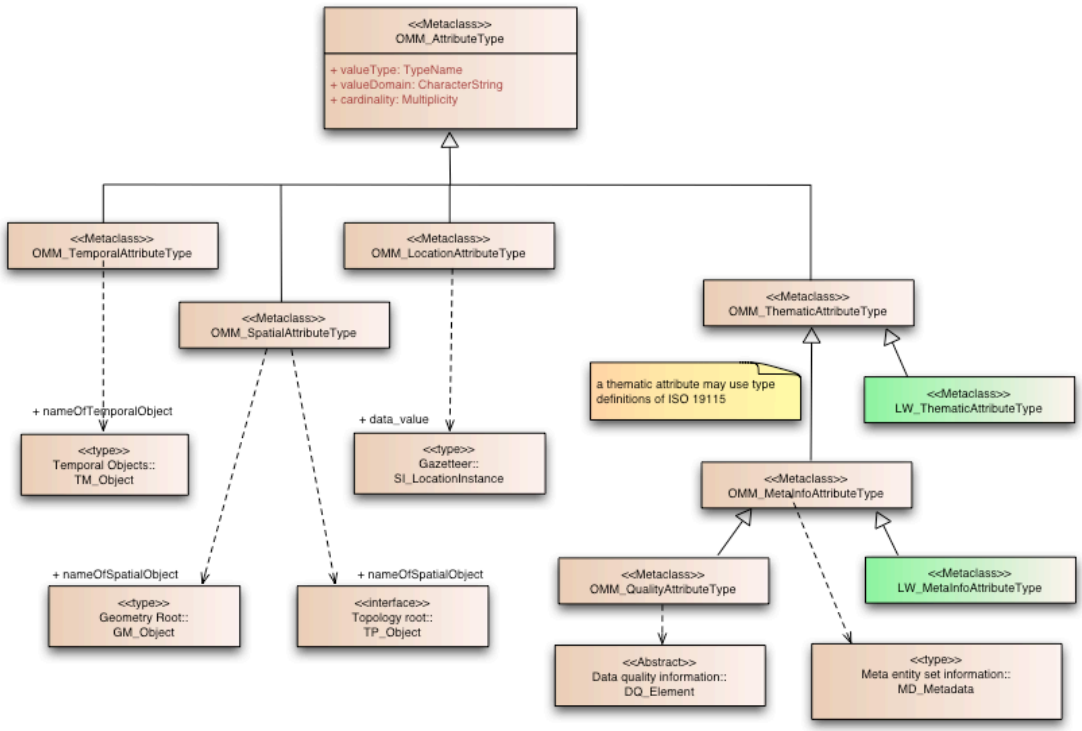


Figure 53: OMM/LW Attribute types (derived from ORCHESTRA RM)

B.2.4 Extensions to feature types

Based on the requirements for the environmental risk management theme for ORCHESTRA, the RM-OA extends the ISO 19109 definition of feature type (GF_Feature_Type). Two categories of information were identified, which can be expressed as special feature types:

- Document Type (for formatted documents, audio and video files, images, etc) and,
- Coverage Type (for features with multiple values for different spatial, temporal or spatiotemporal attributes.

ORCHESTRA recommends keeping the number of predefined feature types very restricted in order not to move to much domain specific information into the meta-model, which might impose too many constraints on application schemas. However, we propose to use these extensions for LifeWatch Reference Model domains due to the following considerations:

- LifeWatch applications should ensure the exchange of heterogeneous biodiversity related data, including multimedia files
- LifeWatch should allow access to the original data, e.g. for provenance purposes. This could be implemented by delivering the data as exchanged by a data provider as a document, which is wrapped in the conceptual model as document type;
- Biodiversity related data often has a spatio, temporal or spatio-temporal reference given by a location, a place name, a modification time stamp or spatio-temporal dependent attributes (for sensor data, itineraries, tracks, etc.).
- A candidate new feature type is Taxon type, since taxonomy is a core sub-discipline involved in LifeWatch

B.2.4.1 Document descriptor type

The Document Descriptor type is a representation of some resource containing information that can be treated as a unit (document). Documents can be grouped according to their MIME media type. Figure 54 specifies the document schema used by ORCHESTRA (see RM-OA V2. Section 8.7.5.2)

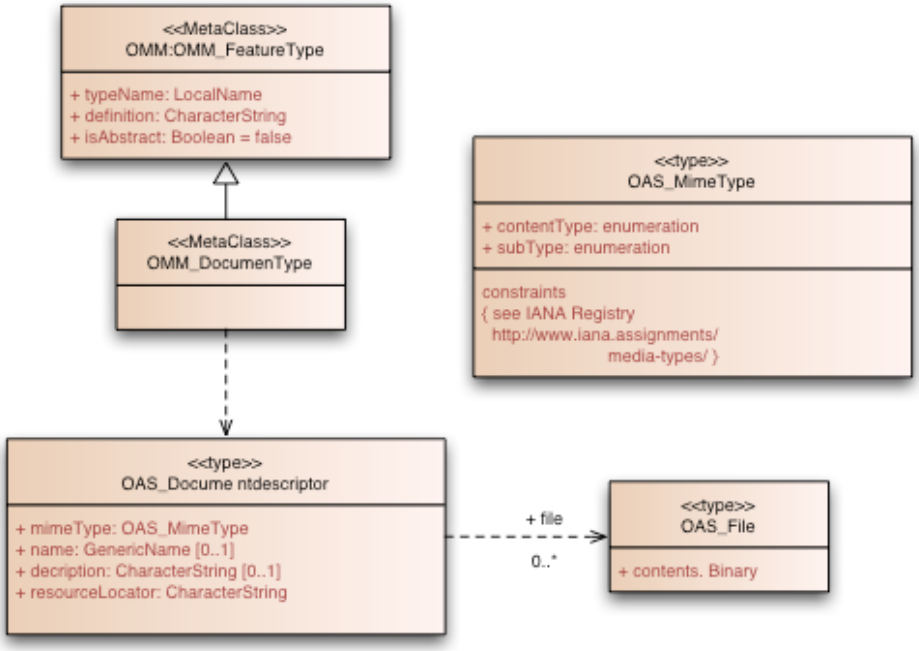


Figure 54: ORCHESTRA schema of the OMM extension "Document Type" (Source: RM-OA)

B.2.4.2 Coverage type

Some 'coverage' represents a continuous geographic phenomenon having no specific extent but expressing a variation (of something) over space, time, or both. A descriptive value for a continuous phenomenon is only meaningful at a particular position in space and possible time. An example is temperature, which has descriptive values for the different locations where it has been measured or interpolated but which exists everywhere within the 'coverage'. In contrast, discrete phenomena characterise recognizable objects that have relatively well-defined boundaries or spatial extent (e.g. a Country, a measurement station, etc.). Both phenomena types are by no means exclusive, since they depend on how they are represented inside a particular application. For example, an ecological region (e.g. a national park) may be viewed as a discrete feature, describing the area and its properties, or as coverage, giving information about species or soil type for each position inside the park and for a temporal range. Moreover, some coverage can be derived from a collection of discrete features with common attributes (e.g. a collection of specimen observations) and vice versa.

The OMM_Coverage type refers to the concept of coverage defined by ISO 19123[141], as a feature, which "acts as a function to return values from its range for any direct position within is spatial, temporal or spatiotemporal domain". Those features have multiple values for each attribute type, where each position within the spatio-temporal representation of the feature has a single value for each attribute type". Examples include raster images, polygon overlays or digital elevation matrix.

Coverage may be discrete or continuous. Discrete coverages have a finite collection of geometric objects and their direct positions: for each geometrical position, a single record of feature attribute values is mapped. An example is the mapping of a set of observation locations with the species found there. The domain of continuous coverages consists of a set of direct positions in a coordinate space: for each geometrical position, different value records are mapped. An example is mathematical functions defining the mapping.

B.2.5 ORCHESTRA's platform-neutral modelling rules for application schemas

ORCHESTRA specified the modelling process for platform neutral application schemas as a set of rules, following the ISO 19109 modelling approach. It allows to derive platform specific application schemas from the conceptual applications schemas automatically and in a normative way.

LifeWatch should follow the ORCHESTRA approach based on a those set of rules defined by ORCHESTRA. It is envisage that LifeWatch will extend the rules, and may be modify them in order to reflect rules of other normative bodies such as, for example, INSPIRE and TDWG.

The application schema level allows a common and correct understanding of the content and structure of data within a particular application field. By mapping the concepts to application schemas it is possible to provide a computer readable schema to be applied for automated mechanisms. Section 8 of ISO 19109 defines the rules to be applied for constructing application schemas. The process described by ISO 19109 refers to a platform-neutral modelling task, since the rules demand that the model is expressed in a conceptual modelling language (e.g. as a conceptual UML model).

ISO 19109 defines four steps for creating an application schema, which are related to figure "Process from reality phenomena to feature instance", derived from ISO 19109. The steps are:

1. Survey requirements from the intended field of application (universe of discourse)
2. Make a conceptual model of the application with concepts defined in the General Feature Model (identify feature types. their properties and constraints)
3. Describe the application schema in a formal modelling language (e.g. UML) according to the rules defined by ISO 19109.
4. Integrate the formal application schema with other standardised schemas (spatial schema, quality schema) into a complete application schema.

The platform-neutral modelling approach allows automatic derivation of platform-specific application schemas, for instance GML Application Schemas, according to ISO 19136. RM-OA defines rules, for:

a) Identification of ORCHESTRA Application Schemas (OAS);
b) Documentation of OAS;
c) Integration of OAS and other schemas;
d) Usage of Types and Stereotypes; and,
e) Specification of an OAS, among others.

---

[141] The coverage model is defined by ISO 19123 and the OGC Abstract Specification Topic 6, Schema for coverage geometry and functions (07-011)

The most relevant characteristics are142:

- The abstract specification of an OAS should use UML 2.0 as its conceptual schema language;
- Each application schema should be identifiable via a name and version. It is recommended to use globally unique names;
- An application schema should be documented according to ISO 19110;
- The data structures of the application shall be modelled in the application schema;
- An application schema can be built up of several other application schemas, which are derived from a standardised application schema, the LifeWatch conceptual schema or a previously defined application schema. The integration of all schemas should be implemented using the dependency mechanism in UML;
- Application schemas for meta-information should use feature attributes according to the Metadata Schema (ISO 19115);
- Descriptions of quality shall be provided in accordance with the requirements of ISO 19113 and ISO 19114. They define data quality elements for quantitative information and data quality overview elements for qualitative information;
- Any description of temporal aspects applied to geographic data shall be in accordance with ISO 19108. Temporal characteristics of a feature type shall be defined as a temporal attribute;
- The domain of spatial attribute types should be in accordance with ISO 19107, which covers spatial data with discrete boundaries, its characteristics and a set of operators to be applied to these concepts;
- When building an application schema associated with a feature catalogue (e.g. an AS-MI for discovery purposes) the information should be in accordance to ISO 19110;
- The value domain of attributes using spatial referencing by geographic identifiers shall be in accordance with the specifications given in ISO 19112, Gazetteer schema;
- A general rule to be applied on deciding, if a concept should be modelled as a feature type or as attribute type is: is the concept of particular importance for the application, has an identity and can be considered as an abstraction of a real world phenomenon, then it should be modelled as a feature type, if the concept does not have an identity, is an auxiliary concept or will be always used in the context of a feature, then it should be modelled as an attribute type.

B.3 The service type meta-model

B.3.1 Service specification rules derived from ORCHESTRA

The LifeWatch meta-model for services extends the ORCHESTRA Meta-Model for Services. For more details with regard to the ORCHESTRA Meta-Model, we refer to RM-OA V2, Section 9.2.

The schema for the specification of Service Types is based on the MetaClass OMM_ServiceType, which is structurally refined by the OMM_InterfaceType. An OMM_ServiceType must have exactly one abstract description (OMM_ServiceAbstractDesc) and zero or more implementation specifications (LMM_ServiceImplSpec). The implementation specifications are attached to a platform by the OMM_PlatformSpec, which provides attributes to specify the platform name, the execution context, the interface language, etc. For interface types zero or more operations could be defined using the meta class OMM_OperationType, which has attributes for name, a Boolean attribute optional, to denote if the operation may be omitted and association attributes to specify parameters for request, and exceptions (see RM-OA V2, Section 9.2).

LifeWatch extends the ORCHESTRA Meta-Model in several points (the changes being indicated by the prefixes LMM and LA). In particular, meta-information for purposes is considered as being part of the

142 See RMOA V2 Rev 2.1 Section 8.8

Abstract Service Description and the Service Implementation Specification. Furthermore, the Service Implementation Specification reflects the OASIS SOA Reference Architecture (see Section 7.4)

Figure 55 illustrates the abstract model for the specification of a Service Type.

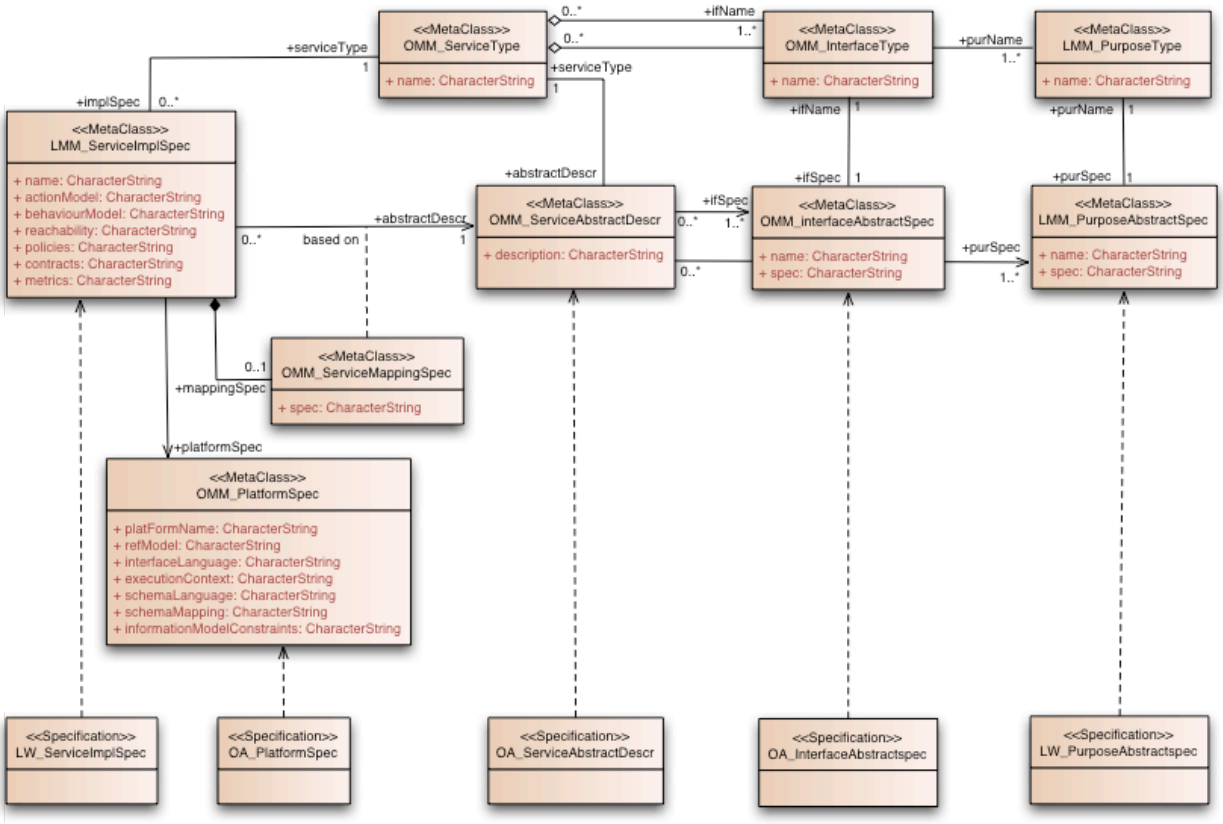


Figure 55: Specification process for LifeWatch Services

The rules for ORCHESTRA service specification (see RM-OA V2. Section 9.2) apply. They can be summarised as follows:

- For all ORCHESTRA Service Types an abstract description (in UML) shall be provided
- An instance of OMM_ServiceType shall be implemented as a CLASS stereotyped as <<ServiceType>>. The name of the class shall correspond to the service type name and shall be uniquely identified for all applications of the ORCHESTRA architecture
- An instance of OMM_ServiceAbstractDesc shall be implemented as a CLASS stereotyped as <<Specification>>. It shall describe the purpose and scope of the service type and shall provide an overview of the interface types supported by the service type and refer to one or more instances of OMM_InterfaceAbstractSpec
- An instance of OMM_InterfaceType shall be implemented as a CLASS stereotyped as <<Interface>> and define the set of operations implemented as instances of OMM_Operation
- An instance of OMM_InterfaceAbstractSpec shall be implemented as a CLASS stereotyped as <<Specification>> and shall provide an overview about the interface type. An instance of OMM_InterfaceType shall be contained in exactly one instance of OMM_InterfaceAbstractSpec.
- An instance of OMM_OperationType shall be implemented as an OPERATION of a class stereotyped as <<Interface>>. The associated request parameters shall be implemented as parameters of the interface operations. The associated result parameters shall be implemented as return types of the interface operations.

- An instance of OMM_RequestParameterType, of OMM_ResultParameterType, or of OMM_ExceptionParameterType shall be implemented as CLASS stereotyped as <<Type>> and shall obey the rules for instances of OMM_AttributeTypes section.
- An instance of OMM_OperationType together with its related instances of parameters shall be implemented as a CLASS stereotyped as <<DataType>>
- For each service that is considered to be available for a given platform an implementation specification for this platform should be available. Instances of OMM_PlatformSpec shall be implemented as CLASS stereotyped as <<Specification>> and shall describe basic properties of the platform.
- A service mapping specification (OMM_ServiceMappingSpec) shall document the mapping of the abstract specification to an implementation specification, and should be a section in the Implementation Specification.
- An Implementation Specification of a Service Type (instance of OMM_ServiceImplSpec) shall be provided according to the rules of the chosen platform. It should be a document structured according to a template that fits the chosen platform.

Note: The typing of various attributes in terms of "CharacterString" is a very mild restriction. An example is the description property of OMM_Constraints. ISO 19109 uses this vague type to allow the expression of constraints in a language specific to the implementation environment of applications. The final LifeWatch Meta-Model for types may decide to enhance the typing in that specific language types are used, e.g. RDFGraph.

Similarly, additional features may be needed. For instance, a feature "formalDescription" may be added at several places being of type "RDF" while the feature "description" remains of type "CharacterString" to hold plain textual descriptions. The adaptation of metamodels is a LifeWatch policy (see Section 7.6).

B.3.2 Service specification rules regarding semantic issues

Editors note: This section is preliminary, a thorough discussion on semantics will be provided in the next version

LifeWatch services will be used in mediation frameworks and should support semantic descriptions. In ORCHESTRA semantic issues are treated at the semantic level, e.g. by considering ontologies defined and shared in user communities (Bügel et. al. 2007)143, lying above the schema level for service types and (meta-) information models.

LifeWatch mediation models can be based on ontologies, data dictionaries, or knowledge-based information systems (Section 7.7.5.2). Based on the assumption that LifeWatch's approach will rely on a core ontology supporting multiple domain ontologies, the following semantic rules can be derived:

- New meta-classes in the Meta-Model level should take into account high level concepts defined at the core ontology or any other community ontology
- Service Meta-information should reference to elements of the core ontology or any other community ontology

143 Bügel, U., Hilbring, D. 2007. Application of Semantic Services in ORCHESTRA. International Symposium on Environmental Software Systems ISSES 2007. Prague, Czech Republic, May 22-25, 2007, available at http://www.eu-orchestra.org/docs/20070522-OrchestraPaper-ISESS2007-ApplicationOfSemanticServicesInORCHESTRA.pdf (November 2009)

- The provision of ontology-based standard schemas (e.g. OWL for Services (OWL-S)144, SAWSDL. or Web Service Modelling Ontology (WSMO)145) for service descriptions (service capabilities) should be supported

This rules may imply that the class LW_ServiceAbstractDesc derived from the meta-class OMM_ServiceAbstractDescription should provide the following information, according to the given standards:

- NonFunctionalProperties (WSMO): The WSMO recommended non-functional properties are Accuracy, Contributor, Coverage, Creator, Date, Description, Financial, Format, Identifier, Language, Network-related QoS, Owner, Performance, Publisher, Relation, Reliability, Rights, Robustness, Scalability, Security, Source, Subject, Title, Transactional, Trust, Type, Version
- Importing ontology (WSMO)
- Mediator (WSMO): Ontology mediators (ooMediator) are used to import ontologies when steps for aligning, merging, and transforming imported ontologies are needed. A Web service can use wwMediators to deal with process and protocol mediation.

144 OWL-S (2004): OWL-S: Web Ontology Language for Services http://www.w3.org/Submission/2004/07/, 2007/02/21
145 WSMO (2006): WSMO: Web Services Modelling Ontology http://www.wsmo.org/, 2007/02/21

B.4 The meta-model for meta-information

B.4.1 Relevant meta-information or meta-data models

Metadata provides context information for data to facilitate the understanding, usage, and management of data by humans or machines. Nevertheless, metadata is a very controversial term, what is data in a context can become metadata for another one and vice versa.

LifeWatch assumes the term meta-information as defined by ORCHESTRA to differentiate between metadata (context data not according to a LifeWatch model) and meta-information, which complies with the purpose-oriented LifeWatch meta-information models. This introductory section states different definitions of metadata, meta-information, metadata elements, and metadata standards from relevant standardisation bodies that should be taken into account on defining the meta-information models for LifeWatch.

Metadata at TDWG

The Biodiversity Information Standards (TDWG) has developed a number of independent approaches to handling metadata describing digital data sets (e.g. as part of the DiGIR protocol and within the ABCD and SDD data standards), but a "common metadata model suitable for describing any biodiversity data set" is still necessary [Hobern2007]146. This standard should facilitate the selection of "those data sets appropriate for a given purpose or application, as well as basic information on provenance, ownership, and intellectual property" [Hobern2007]. Besides, the data model should provide technical metadata (e.g. on how to access the data, how to comply with secure access mechanisms) and include taxonomic, geographic, and temporal dimensions on the data set and the collecting methods.

The LifeWatch infrastructure seeks to find or define such common models in collaboration with standardisation bodies.

ORCHESTRA Meta-Information Models

The ORCHESTRA Framework for Meta-Information Models suggests the following approach for the development of meta-information models (RM-OA V2 Section 8.4.1):

- Find the purposes (use cases/functions) in the context of users and/or machines.
- Develop the meta-information model(s) for data and/or services in this respective context.
- Specify the Meta-Information model for the application Schema (OAS-MI) based in the ORCHESTRA meta-information rules.

Although purposes are often application specific, some of them can be abstracted from generic usage goals. ORCHESTRA RM-OA defines a set of rules for "well-known particular purposes" for the use of metadata. Those are: a) Discovery, b) Access, Storage and Invocation, c) Integration, d) Interpretation, e) User Profiling, f) Authentication, Authorisation and Accounting (AAA), g) Quality control, h) Transaction Management and i) Configuration Management".

The ORCHESTRA RM-OA does not mandate the usage of one particular meta-information model, but gives the designer the freedom of specifying a meta-information model for various purposes according to the rules of the meta-model.

---

146 [Hobern2007] Hoberm, Donald 2007: "The New Chairman' Vision for 2009, 06-Nov-2007, available at http://www.tdwg.org/about-tdwg/news/article/the-new-chairmans-vision-for-2008/

EML Levels of Completeness

Similarly to the ORCHESTRA approach the ILTER network produced a set of best practice recommendations for the use of Ecological metadata Language (EML), which recognised 6 levels of "completeness" for a prospective use of data147. Those are i) Identification, ii) Discovery, iii) Evaluation, iv) Access, v) Integration, and vi) Semantic Use. GBIF's strategic plan intends to collect metadata to cover the first four levels. The goal of LifeWatch is to provide capabilities through its infrastructure for all the EML-levels, in particular for the last two levels, namely integration and semantic use.

INSPIRE Metadata Regulation

The INSPIRE regulation for metadata148 establishes that a set of compatible and usable metadata elements for describing spatial data sets and services in the European Community should be created and maintained. The Metadata Implementing Rules document149 defines how metadata models for spatial related data can be implemented based on EN ISO 19115 and EN ISO 19119 and ISO 15836 (Dublin Core). Metadata elements are the smallest unit for a metadata property: they refer to classes, attributes, or relationships between components in a UML diagram.

ISO 19115 – Geographic information -- Metadata

ISO 19115 defines the schema for the identification, extent, quality, spatial and temporal schema, spatial reference, and distribution of digital geographic data. These schemas are useful for the cataloguing of datasets, clearinghouse activities, and the full description of datasets; geographic datasets, dataset series, and individual geographic features and feature properties. ISO 19115 defines more than 300 metadata elements, most of which can be applied optionally. Mandatory elements are defined as "core" metadata. ISO 19115 defines 10 "core" metadata elements. ISO/TS 19139:2007 is the XML schema implementation of ISO 19115 as GML encoding. Applications can create profiles and add new elements to the defined by ISO 19115. The

The description of ISO 19115 metadata elements contains the following information:

- Name: unique label
- Short name and domain code:
- Definition: description of the metadata element
- Obligation: Indicator whether a metadata element shall always be documented or not: (M) = Mandatory, (C) = Conditional (at least one entity is mandatory under specified restrictions), and (O) = Optional
- Condition: specifies the condition for conditional elements
- Maximum occurrences: Maximum number of possible instances
- Data type: Set of distinct values for representing the element
- Domain: The allowed values or the use of free text

ISO 19115 presents metadata for geographic data in Metadata Sections, which are represented as UML150 packages. Each metadata section (package) contains one or more Metadata Entities, represented as UML classes

147 http://knb.ecoinformatics.org/emlbestpractices_oct2004.doc

148 See and Commission Regulation No 1205/2008 implementing Directive as regards metadata

149 Metadata Implementing Rules: Technical Guidelines based on ISO 19115 and EN ISO 19119 http://inspire.jrc.ec.europa.eu/reports/ImplementingRules/metadata/MD_IR_and_ISO_20090218.pdf

150 Unified Modeling Language (UML): a OMG standard for general-purpose modeling of software systems (structure, behaviour, architecture, business processes, and data structure) http://www.omg.org/UML/#UML2.0

OGC and FGDC

ISO 19115 was adopted as a replacement for OGC Abstract Specification Topics 9 and 11. FGDC in conjunction with ANSI INCITS L1 are planning the migration of the FGDC Content Standard for Geospatial Metadata to be a profile of ISO 19115.

The OGC Abstract Specification Topic 12 - The OpenGIS Service Architecture , also published as ISO 19119:2005, defines a service metadata schema for use in a catalogue service as is done for dataset metadata.

The OGC ISO Metadata Application Profile of the OGC Catalogue Specification defines an application profile of the CSW for ISO 19115/ISO 19119 metadata with support for XML encoding per ISO/TS19139. This application profile specifies the interfaces, bindings, and encodings required to publish and access catalogues of metadata for geospatial data, services, and applications using the ISO 19115 and ISO 19119 standards.

The OGC CSW-ebRIM profile of the OGC Catalogue Specification defines means to customize an OGC catalogue service using the OASIS ebXML registry information model (ebRIM). Using the profile, a catalogue service can be adapted to meet the needs of a community within the geospatial domain. For example, a "Portrayal" package might include elements for working with the style descriptors and symbol collections used in map production. A "Geodesy" package can include elements for defining coordinate reference systems and related components such as a datum and a prime meridian. A CSW-ebRIM package for ISO 19115 and ISO 19119 is under development.

The FGDC Biological Data Profile (BDP) is an approved profile to the FGDC-Content Standard for Digital Geospatial Metadata (CSDGM) proving additional fields to the FGDC-CSDGM standard that allow biological information such as taxonomy, methodology, and analytical tools to be added to a metadata record. The CDSGM is also grouped in Sections as ISO 19115. The biological extensions are:

- Taxonomy Data (section Identification Information)
- Description of the Geographic Extent and Bounding Altitudes in Spatial Domain (section Identification Information)
- Analytical Tool (section Identification Information)
- ASCII File Structure (section Distribution Information)
- Methodology in Lineage (section Data Quality Information)
- Geologic Age (section Time Period Information)

Dublin Core Metadata Initiative (DCMI) / ISO 15836:2009

The Dublin Core Metadata Initiative specifies DCMI Metadata Terns, containing the Dublin Core Metadata Element Set, which has been standardized as ISO Standard 18536:2009 and NISO Standard Z39.85-2007. Dublin Core provides a “universal” description of resources. LifeWatch should support to export metadata as a Simple Dublin Core Metadata Element Set, which consist in the following 15 elements: (1.) Title, (2.) Creator, (3.) Subject, (4.) Description, (5.) Publisher, (6.) Contributor, (7.) Date, (8.) Type, (9.) Format, (10.) Identifier, (11.) Source, (12.) Language, (13.) Relation, (14.) Coverage, and (15.) Rights, according to the Dublin Core Abstract Model151

[151] http://dublincore.org/documents/abstract-model/

OpenSearch

OpenSearch is a de-facto standard for a collection of simple formats extending existing schemas such as ATOM and RSS and covering discovery and description documents for search engines. The main idea is that search results are delivered as web feeds. OpenSearch provides enough semantics to express full-text searches open search description document. The main documents are:

- Description documents for search engines
- Template URLs that describe how to invoke a search
- Paged search results
- Auto-Discovery of Description documents

The OpenSearch specification is made available under the Creative Commons Attribution-Sharealike 2.5 license. In addition, the OASIS Search Web Services group is publishing an Abstract Protocol Definition of the interface or "binding", which coincides with the community specification published at http://opensearch.org. The OGC Members approved OpenSearch Geospatial Extensions as an OGC Discussion Paper.

LifeWatch approach

For the LifeWatch infrastructure, the defined purposes should be based on ORCHESTRA purposes and ILTER levels of compliances, adding capabilities extracted from the biodiversity capabilities of LifeWatch, as needed.

The following LifeWatch purposes are outlined in the Section B.5.4:

1. Discovery
2. Identification
3. Access, Storage and Invocation
4. Integration
5. Orchestration
6. Mediation
7. Personalisation
8. Quality evaluation
9. Provenance
10. Authentication, Authorisation and Accounting (AAA)
11. Data capture (sensors and mobile devices)
12. Collaboration
13. Human Interaction

One should note that this list is not exhaustive. As stated above, it covers well-known generic purposes essentially related to basically any system architecture. In LifeWatch the need for more, particularly "semantically based" purposes may arise. For instance, one may imagine a purpose "taxonomy" or "taxon occurrence record" that covers all the meta-information needed to deal with taxonomic names or taxon occurrence records. On the other hand, such data may as well be captured using a feature type. Both are valid design decisions. In case of taxon occurrence records for instance, using a meta-information model may be adequate in that may cover the base information common to all related data models such as ABCD or DarwinCore.

The LifeWatch purpose-oriented information models should be constructed in order to comply with the following standards and profiles, e.g. through providing wrappers, translation mechanisms, or including the elements into the schemas to support the standards:

- ISO 19115 – Metadata for Geographic Data
- ISO 19115-2 – Geographic Information Metadata Part 2: Extensions for imagery and gridded data
- ISO 19119 – Metadata for services
- INSPIRE profile of ISO 19115 and ISO 19119
- ISO 19115 Profile for CS-W (OGC)
- ISO 15836 (Dublin core)
- FGDC CSDGM Application Profile for CSW (Best Practice OGC Document) and FGDC Biological Data Profile
- OpenSearch Specification152, due to its current relevance to World Wide Web searching.

B.5 The LifeWatch Conceptual Model for Meta Information

B.5.1 Introduction

The LifeWatch Conceptual Model for Meta Information provides, on one hand, the basic elements for building abstract meta-information models for each purpose. On the other hand, the model provides rules on how to construct meta-information models and it defines generic purposes for LifeWatch applications, giving examples of abstract meta-information models for them. Therefore, it is considered a conceptual model, since it provides an abstract description of elements and rules from which purpose-oriented and application specific meta-information models can be derived.

The conceptual model for meta-information will primarily follow the principles and rules of ISO 19115, facilitating the building of meta-information models compliant to INSPIRE, ORCHESTRA, OGC, and FGDC. The basic elements are the basic concepts of LifeWatch meta-information and will be described using object-oriented terms, facilitating the construction of UML diagrams: concepts are describes as Classes, relationships between concepts as Associations, and Packages are containers grouping related classes and association to describe a particular. Packages provide a common namespaces for the contained elements. Packages may contain other packages, allowing building hierarchical structures. LifeWatch's conceptual model for meta-information is, in concrete, a profile of the ISO 19115 and ISO 19119 standards and a set of rules based on the rules for ORCHESTRA application schemas for meta-information (summarized in Appendix B.2.5) and on the INSPIRE implementing rules (Appendix B.1.1).

ISO 19115:2003 Annex 3 groups metadata elements into the Packages depicted in Figure 56. LifeWatch shall use the ISO 19115 packages as standard packages, adding further elements, if necessary, to comply with the Standards mentioned in Appendix B.1. The package elements are described in tabular form, indicating the origin source of the element in the column Standard. For possible LifeWatch specific extensions or constraints LifeWatch is used as Standard entry.

The LifeWatch profile of ISO 19115 and ISO 19119 will serve as basis for the development of purpose-oriented meta-information schemas.

152 http://www.opensearch.org/

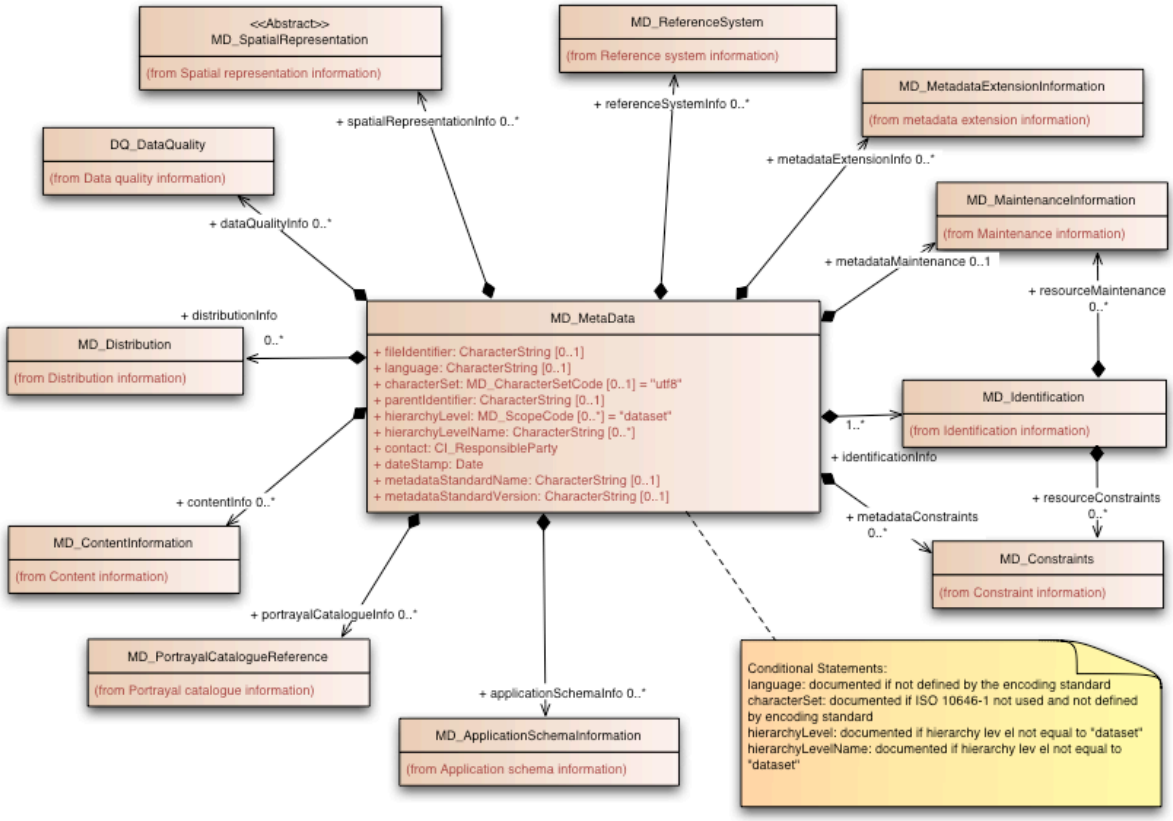


Figure 56: Metadata packages according to ISO 19115:2003 Annex A

The tables in the next sections contain the following information:

- Metadata Element - Gives the name of the element, preferable as defined on a standard. The cardinality is given in square brackets ([]). For example [0..n] refers to an optional element, which can have up to n different values. Cardinalities beginning with a number bigger than 0 express mandatory elements. Alternative cardinalities, indicated by slashes (/) express conditionality, depending on a particular case explained in the description column.
- Data Type - Identifies the data type and domain of the element. It may refer to a basic type or to a predefined type. Predefined types are normally described through a basic type, given behind a colon (:) or a set of attributes, written into braces ({ })
- Standard - Refers to the original standard defining the meta-data element. If the element is a recommended LifeWatch extension, then the entry will be LifeWatch.
- Corresponds to -A reference to equivalent elements in other relevant standards is given to facilitate the translation and thus compatibility between the models. The following version of standards were taken into account:
    - EML V 2.1.0 (XML)
    - Dublin Core Metadata Element Set V 1.1 (RDF)
- Description - Provides the meaning of the element and additional comments, e.g. for conditionality restrictions or comments from the extension of a standard.

B.5.2 Standard packages

B.5.2.1 Metadata information (MD_Metadata package)

For all metadata-information models, metadata about the meta-information should be stored. Thus, the package "MD_Metadata" is the basis package for all models and should contain the elements shown in Table 4 below.

Table 4: Elements of the package: MD_Metadata

Metadata Element	Data Type	Standard	Description
fileIdentifier [0/1]	MD_Identifier:String	ISO 19115	Unique identifier for the metadata file, mandatory for datasets and dataset series
language[0/1]	LanguageCode (ISO/TS 19139)	ISO 19115	Language used within the metadata.
characterSet [0..1]	CodeList ISO19115 B.5.10	ISO 19115	Character encoding standard used for the metadata set.
hierarchyLevel [1..n]	MD_ScopeCode:CodeList ISO 19115 B.5.25	ISO 19115	Scope to which the metadata applies. E.g. "service", "dataset", "series"
hierarchyLevelName [0..1]	String	ISO 19115	Name of the hierarchy level
contact [1..n]	CI_ResponsibleParty: {organisationName, contactInfo, role :CI_Rolecode}	ISO 19115	Point of contact for the metadata. Each metadata use for a given purpose should give a reference of responsibility. If the metadata has been semi-automatically extracted, the organisation using the extraction service should be named.
dateStamp [1]	Date	ISO 19115	Creation date for the metadata
resourceIdentifier	MD_Identifier	ISO 19115	Identifier of resource to which the metadata applies.
metadaStandardName [0..1]	String	ISO 19115	Name of standard or profile name used
metadataStandardVersion [0..1]	String	ISO 19115	Version of standard or profile used

B.5.2.2 Identification information (MD_Identification)

Table 5 refers to base properties applicable for all resource types. Further properties are grouped into those are relevant for dataset and series resources (MD_DataIdentification) and those relevant for service resources (MD_ServiceIdentification).

Table 5: Elements of the package: MD_Identification

Metadata Element	Data Type	Standard	Description
Citation [1]	CI_Citation: {title, alternateTitle,	ISO 19115	Citation data for the resource. The identifier should be a unique identifier. The

Metadata Element	Data Type	Standard	Description
	date, edition, editionDate, identifier, citedResponsibleParty, presentationForm, series, otherCitationDetails, collectiveTitle}		citedResponsibleParty is the author of the Resource. Date of publication should be mandatory. CitedResponsibleParty should be mandatory.
abstract [1]	String	ISO 19115	Summary of the content (Textual)
pointOfContact [1..*]	CI_ResponsibleParty	ISO 19115	Person or institution who maintains the data, it should be at least on point of contact
descriptiveKeywords [1..*]	MD_Keywords_PropertyType: {keyword, type, thesaurusName}	ISO 19115	Provide category keywords, their type and reference source. Should be mandatory for at least one keyword. The keyword CodeList should be adjusted for LifeWatch.
resourceFormat [0..*]	MD_Format: CodeList	ISO 19115	Format of the resource. For services protocols and for resources data formats that are supported
resourceConstraints [0..*]	MD_Constraints_PropertyType: {useLimitation: String] }	ISO 19115	Constraints for the usage of the resource. Constraints categories are defined for legal, security issues. The description of useLimitation should be specified in a semantic language
aggregationInfo [0..*]	MD_AggrefateInformation_PropertyType: {aggregateDataSetName: CI_Citation, aggregateDataSetIdentifier, associationType:CodeList, initiativeType: CodeList}	ISO 19115	Information about how the data is been aggregated. Only applicable for datasets. LifeWatch should adjust CodeList for associationType and initiativeType
Version	String	LifeWatch	Possible LifeWatch extension for provenance: just if the version of the dataset, series, or service is the same, the reproducibility can be guaranteed

B.5.2.3 Data Identification Information (MD_DataIdentification)

Table 6: Elements of the package: MD_DataIdentification

Metadata Element	Data Type	Standard	Description
spatialRepresentationType [0..*]	CodeList	ISO 19115:2003	Type of spatial representation of the data set, e.g. grid or vector
spatialResolution [0..*]	MD_Resolution_Type : {equivalentScale, distance}	ISO 19115:2003	Resolution value for raster data given as a scale denominator (e.g. 50000 for 1:50,000) or distance value and a unit of

			measures for the ground sample distance
Language [0.*]	CodeList	ISO 19115:2003	Language of the dataset or data series
characterSet [0..*]	CodeList	ISO 19115:200	Character set applicable to the dataset or series
topicCategory [1..*]	MD_TopicCategoryCode_PropertyType : {	ISO 19115:2003	Classification for a topic
extent [1..*]	EX_Extent_PropertyType: {description: String, geographicElement: EX_GeographicExtent, temporalElement: EX_TemporalExtent, verticalElement : EX_VerticalExtent. taxonomicExtent: LW_taxonomicExtent}	ISO 19115:2003	Scope of the dataset. For LifeWatch at least one spatial extent is mandatory (also for INSPIRE). LifeWatch extends the extent property to add taxonomic information for biodiversity data, and add further properties to the geographical extent. The extent types are explained in more detail below.
supplementalInformation [0/1]	String	ISO 19115:2003	Additional text, which can be used for semantic interpretation

Figure 57 shows the ISO 19115:2003 schema for the EX_Extent data type. A short description of the syntax is given below.

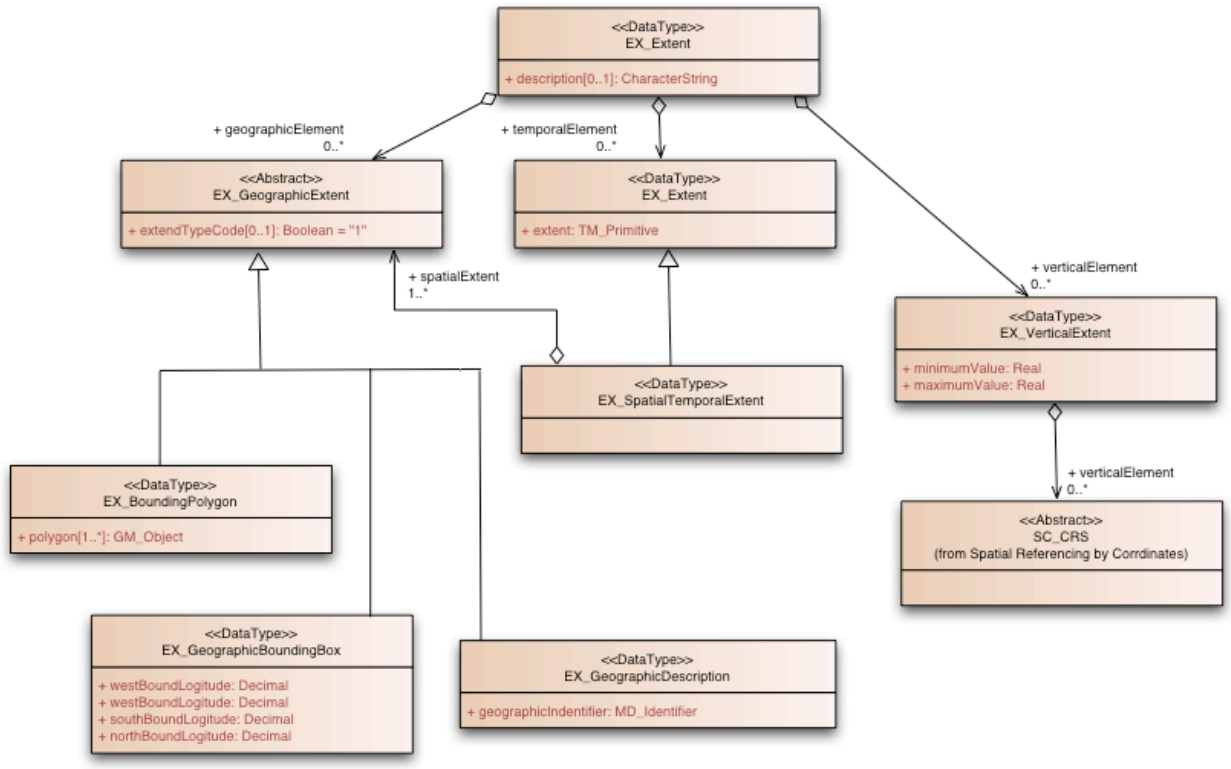


Figure 57: Hierarchy of ISO 19115:2003 EX_Extent

EX_GeographicExtent:

Geographic extents describe the area of the entire dataset. ISO 19115 specifies the description through either a geographic bounding box, defined through the coordinates, given in the WGS84 coordinate reference system (CRS), for the smallest rectangular are containing all the data. INSPIRE relax the CRS constraint by allowing any geographic coordinate reference system with a Greenwich Prime Median. This should be applicable for LifeWatch too. The coordinate reference system (CRS) should be given, where WGS84 remains the default CRS. More than one BoundingBox are allowed, from which a maximum BoundingBox can be generated if required, e.g. for Web Map Services. LifeWatch should use the possibility to define the extent by a geometric object, e.g. a point, a polygon, or a set of polygons, allowing more precisely descriptions of the extent. This is foreseen in ISO with the BoundingPolygone type, whereby the geometry must not be a polygon. The other possibility to express a geographic reference is by giving a geographic description (e.g. a geographic name). A BoundingBox can be generated from the geometry or from the location name (e.g. through a gazetteer) if needed. LifeWatch extends the ISO 19115:2003 EX_GeographicExtent element and therefore the data type prefix is changed from EX_ to LifeWatch:

```
LW_GeographicExtent [1..*]: {
LW_GeographicBoundingBox [0..*]: {CRS, westBoundLongitude, eastBoundLongitude,
                                  southBoundLatitude, northBoundLatitude}
EX_BoundingPolygon [0..*] : {polygone: GM_Object
EX_GeographicDescription [0..*] :{geographicIdentifier: MD_Identifier }
}
```

TemporalExtent

Temporal extents describe the temporal scope of the data set. It includes a type for spatial-temporal extent, which can be applied for coverages.

```
EX_TemporalExtent : { extent: TM_Primitive_PropertyType,
                    spatialExtent :     EX_GeographicExtent }
```

VerticalExtent

Vertical extend refers to the vertical domain of the dataset

```
EX_VerticalExtent: {minimumValue: Real, maximumValue: Real,
                    verticalCRS: SC_CRS_Property}
```

TaxonomicExtent

Taxonomic extent refers to the taxonomic classification of a specie or observation. It should be defined within LifeWatch.

B.5.2.4 Service Identification Information: MD_ServiceIdentification

Table 7: Elements of the package: MD_ServiceIdentification

Metadata Element	Data Type	Standard	Description
serviceType [1]	ServiceCategory (enumeration to be defined)	ISO 19119	Not applicable for datasets. Mandatory for service. Service type name from a registry of services
serviceTypeVersion [1..*]	String	LifeWatch	For searches based on the service type version, e.g. WMS 1.1, etc. It differs from the resource version, which is inherent to the

			implementation of the service.
couplingType[1]	SV_CouplingType	ISO 19119	Mandatory for services. Type of resource coupling; it can be loose, tight or mixed
operatesOn [0..*]	MD_DataIdentification	ISO 19119	Resource id for datasets on which the services operates. Reference to the resources on which it operates.
containsOperations [1..*] -	SV_OperationMetadata : { operationName[1]: String, DCP [1..*] : DCP List, connectPoint [1..*]: CI_OnlineResource, Linkage: URL}	ISO 19119	DCP: Distributed Computer Platform e.g. XML, CORBA, JAVA, COM, SQL, WebServices (Default), Possible extension for LifeWatch OGCWebServices, OSGAServices, ORCHESTRAWebService.
extent [1..*]	EX_Extent (see section B.5.2.3)	INSPIRE	At least one bounding box is required

B.5.2.5 Distribution information: MD_Distribution

Table 8: Elements of the package: MD_Distribution

Metadata Element	Data Type	Standard	Description
distributionFormat [0..*]	MD_Format	ISO 19115:2003	Format, of how the resource can be obtained
Distributor [0..*]	MD_Distributor : {distributorContact, distributionOrderProcess, distributorFormat, distributorTransferOptions }	ISO 19115:2003	Information about distribution for one distributor instance (provider).
transferOptions [0..*]	MD_DigitalTranserOptions_PropertyType : { unitsOfDistribution, transferSize, online, offLine}	ISO 19115:2003	Technical means and media by which the dataset is obtained from the provider.

B.5.2.6 Data quality: MD_DataQuality

NOTE: Data quality elements will be needed both at the dataset level and at the level of individual records. INSPIRE states that there shall be one and only one set of quality information scoped to the full resource and having a lineage statement.

Editor's Note: LifeWatch should evaluate the provenance requirements in order to extend this package.

Table 9: Elements of the package: MD_DataQuality

Metadata Element	Data Type	Standard	Description
lineage [0/1]	LI_Lineage_propertyType: {statement, processStep, source}	ISO 19115	General explanation of the data producer's knowledge about the lineage (derivation) of a dataset. Mandatory for datasets, not applicable for services. Just one set of quality information should be scoped to the full resource
report[0..*]	DQ_Element : { nameOfMeasure, measureIdentification, measureDescription, evaluationMethodType, evaluationMethodDescription, evaluationProcedure, dateTime, result }	ISO 19115	Evaluation report for quality measurements.

scope[1]	DQ_Scope : { level, extent, levelDescrip-tion}	ISO 19115	Scope of quality measurement

B.5.3 LifeWatch specific packages

ISO 19115:2003 Annex 3 groups metadata elements into Packages (see Figure 56). The next sections give an overview of relevant metadata-elements for LifeWatch, extracted from the standard references mentioned above (referred on the column Standard) grouped by the most relevant packages. Possible extensions or constraints for LifeWatch are written in bold italic.

The tables in the next sections contain the following information:

- Metadata Element - Name as defined on a Standard and cardinality is given in square brackets ([]). Alternative cardinalities, indicated by slashes (/) express conditionality which is explained in the description column
- Data Type - Data type and domain of the element. It may refer to a basic type or to a predefined type. Predefined types are normally described through a basic type, given behind a colon (:) or a set of attributes, written into braces ({ })
- Standard - Related standard, which defined the metadata-element. If is a possible LifeWatch extension, then the entry will be LifeWatch.
- Description - Definition of the element and additional comments, e.g. for conditionality restrictions or comments from the extension of a Standard.

Editors Note: *In this version (v0.2) of the present document, we only indicate how these packages may be used in the definition of the LifeWatch meta-information models. The list of purposes and respective meta-information models will be developed in future versions of this document.*

The following Packages are candidates for extending the ISO packages. They are partly derived from OGC models.

B.5.3.1 Service invocation

Table 10: Elements of the LifeWatch extension package: LW_ServiceInvocation

Metadata Element	Data Type	Standard	Description
- operationName[1] () - connectPoint [1..*] CI_OnlineResource ..	String	ISO 19119:2005/Amd.1:2008	Unique identifier for the interface. Describes in ISO Package SV_OperationMetadata.
DCP[1..*]	DCP List	ISO 19119:2005/Amd.1:2008	DCP: Distributed Computer Platform e.g. XML, CORBA, JAVA, COM, SQL, WebServices (Default) as defined in ISO Package SV_ OperationMetadata., Should be extended with OGCWebServices, OSGAServices, LifeWatchServices, etc.
operationDescrip-tion[0..1]	String	ISO 19119:2005/Amd.1:2008	Free text describing the operation, describes in ISO Package SV_ OperationMetadata.
operationType	CodeList	LifeWatch	Classification of operations to allow the searching for

Metadata Element	Data Type	Standard	Description
	(LifeWatch?)		services providing certain operations e.g. modelling, etc
InvocationName [0..1]		ISO 19119:2005/Amd.1:2008	Name used to invoke the interface within the context of the DCP. Describes in ISO Package SV_ OperationMetadata.
parameter [0..*] - Name (1) (MemerName) - Direction (0..1) (SV_ParameterDirection) - description (0..1) - optionality (1) (String) - repeatability(1) (Boolean)	SV_Parameter	ISO 19119:2005/Amd.1:2008	Parameters required for this interface - Name as used by the service - Indication of usage (in, out, In/out) - Explanation of role - Indication if it is required (“Mandatory”/”Optional”) - Indication if more than a value may be provided
dependsOn [0..1] - operationName[0..*]	Set {operationName}	Derived from ISO 19119:2005/Amd.1:2008	List of operations (for the same service) that must be completed before the current operation is invoked

B.5.3.2 Search

Table 11: Elements of the LifeWatch extension package: LW_Search

Metadata Element	Data Type	Standard	Description
query [1..*]	String	ORCHESTRA	Definition of the query. Might be derived from ORCHESTRA Catalogue Service abstract specification for OA_Query (Search Interface), type OA_Query.
searchType	CodeList	ORCHESTRA	Search process (e.g. full text search, geospatial search, temporal search). Might be derived from ORCHESTRA Catalogue Service abstract specification for OA_SearchRequest (Search Interface), type OA_SearchType.
queryLanguage[0..*]	CodeList	ORCHESTRA	Reference to a query language that can be used. The query language can be a well-known language or a user defined language, which specified query parameters and returnable properties. Might be derived from ORCHESTRA Catalogue Service abstract specification for OA_Query (Search Interface), type OA_QueryLanguage.

B.5.3.3 Navigation

Table 12: Elements of the LifeWatch extension package: LW_Navigation

Metadata Element	Data Type	Standard	Description
numberFittingResource	Number	ORCHESTRA	Number of resources, that were found on a search query. Might be derived from ORCHESTRA Catalogue Service abstract specification for OA_SearchResponse (Search Interface) (result count).
resultIndex	Number	ORCHESTRA	Index of the first resource on a result set to be returned as query result. Might be derived from ORCHESTRA Catalogue Service abstract specification for OA_SearchResponse (Search Interface) (cursorPosition)
resultNumber	Number	LifeWatch	Number of resources to be submitted as query result

B.5.3.4 Transaction

Metadata for transaction purposes are not yet defined and could contain the elements in Table 13.

Table 13: Elements of the LifeWatch extension package: LW_Transaction

Metadata Element	Data Type	Standard	Description
transactionType	CodeList	LifeWatch	Action name: update, insert, delete
dateStamp	CI_Date	ISO 19115:2003	Date and time of transaction action. Derived from ISO 19115 MD_Metadata
pointOfContact[0..*]	CI_ResponsibleParty	ISO 19115:2003	Reference to users which performed the transaction. Derived from ISO 19115 MD_Identification
query	String	See Table 11	Transaction query
queryLanguage	CodeList	See Table 11	Dictionary or reference to the language of query

B.5.3.5 Harvesting

Meta-information about the automatic retrieval of resource meta-information for discovery purposes performed by catalogues and registries may contain the following elements, according to the definition of the OpenGIS Catalogue service shown in Table 14.

Table 14: Elements of the LifeWatch extension package: LW_Harvesting

Metadata Element	Data Type	Standard	Description
request	String	OGC CSW/ORCHESTRA	Request query. Could be a fixed value. OGC CSW., Harvest operation request. Might also be derived from ORCHESTRA Catalogue Service abstract specification for

			OA_CollectionMetaInformationRequest (Collection Interface) (operationRequest).
service	CodeList	OGC CSW	Harvesting service performing the action; the service names for harvesting should be known. Request query. Could be a fixed value. OGC CatalogueService Spec., Harvest operation request.
version	String	OGC CSW	Version of harvesting service
namespace	List<String>	OGC CSW	All namespace used by all qualified names in the request
sourceReference	URI	OGC CSW/ ORCHESTRA	Refernce to a source of meta-information to be harvested. Might be derived from ORCHESTRA Catalogue Service abstract specification for OA_CollectionMetaInformationRequest (Collection Interface) (sourceReference).
resourceType	CodeList	OGC CSW/Orchestra	Reference to the resource type to be harvested (see serviceTypes or dataset categories on the services and dataset packages). Might be derived from ORCHESTRA Catalogue Service abstract specification for OA_CollectionMetaInformationRequest (Collection Interface) (collectType).
resourceFormat	String	OGC CSW	Type, e.g. MIME type, indicating format of the resource to be harvested
responseHandler	URL	OGC CSW	Reference to person or entity to which to respond when the asynchronous harvest process has been completed
harvestInterval	Period (ISO 8601)	OGC CSW	Interval for performing the harvest. If not specified then the harvest will be done only once in response to the request

B.5.3.6 Annotations

User may include annotation and comments to the resource, which can serve to give a quality statement on datasets or services. Annotation elements can also serve for provenance.

Table 15: Elements of the LifeWatch extension package: LW_Annotations

Metadata Element	Data Type	Standard	Description
pointOfContact [1..n]	CI_ResponsibleParty	ISO 19115 MD_Identification	Who made the annotation and which role it has
dateStamp [1]	CI_Date	ISO 19115 MD_Metadata	Annotation date
resourceIdentifier[1]	MD_Identifier:String	ISO 19115 MD_Metadata	Resource for which the annotation has been given
annotationType[1]	CodeList	LifeWatch	LifeWatch lists possible entries ("comment", "update", "usage")
value[0..1]	String	LifeWatch	Annotation content
document[0..*]	Document	LifeWatch	Documents containing annotation content.

B.5.3.7 Portrayal

The package contains elements, which can contribute to harmonise the portrayal across different view services. Candidate elements can be extracted from the OGC standard for symbol encoding and from the OGC Web Map Service Interface.

B.5.3.8 Licensing and fees

According to INSPIRE Implementation Rules for Data and Service Sharing (D4.9) licence types should be defined and models for the use of the licences should be provided. A specification example is the OGC Geospatial Digital Rights Management Reference Model (GeoDRM_RM)153.

B.5.3.9 Multidimensional model

Multidimensional models are needed for integration of data at different levels of detail. Possible candidate elements can be extracted from on-line analytical processing technologies like hierarchy, dimension, attribute, etc.

B.5.3.10 Process

The process package should contain relevant information related to the execution of a process like status, execution time, response handler, origin, etc.

B.5.3.11 Messaging

Metadata elements needed for notification and event handling.

B.5.3.12 Sensor

Metadata for handling sensor data. Candidate elements can be taken from the OGC SensorModellingLanguage Specification or from the S@ny project.

B.5.3.13 User profile

Package containing elements related to user preferences like language, colour schemas, membership to user groups, etc

B.5.3.14 User

Package containing elements for user management as user groups, principals, subjects, collaborative communities, user-roles, etc.

B.5.4 Purposes and Meta-Information Models

Candidates for LifeWatch purposes will be outlined below. For the specific purpose “Discovery”, a Meta-Information Model is provided as an example for using the Metadata Packages listed above.

---

153 http://www.opengeospatial.org/standards/as/geodrmrm

B.5.4.1 Purpose "Discovery"

For the purpose "discovery", relevant resources (data, services, or technical equipment) should be traceable according to a set of descriptive attributes. The meta-information model for the purpose "discovery" consists of these attributes. These attributes should, for instance, allow searching by defining constraints parameters like the geographical extension and the time span of the last modification. According to the entity searched, other specific parameters and relations to them will be relevant, like the taxonomic name for species or the service type for services.

These elements are described at the meta-information model of the application schema.

Dictionaries can be used to facilitate the search, allowing the conversion of user entries to taxonomic correct names, (e.g. through syntax-check, thesauri, checklists, taxonomy catalogues) or the transformation of entries to valid data types like the conversion of place names and addresses to coordinates or to the delimited bounding box through gazetteers or geocoding algorithms. As well the inclusion of translation mechanisms ca be used to provide multi-language.

Topic maps that allow an associative approach to the concept searched can help to find things even when one does not exactly know how to call them. The search should also be possible for keywords, categories and for a full-text scan.

Once a search query has been performed, the results should also be navigable, allowing to specify the number and the sorting of the submitted result elements as well as browsing the results with typical commands like “next”, “last”, “next x elements”, etc.

Searching can be an iterative process combining all those mechanism in several loops, where drill into can be very important at specific unclear points.

NOTE: Examples of services based on discovery meta-information models are the OGC Catalogue Service for the Web (CS-W), the ORCHESTRA specification for Catalogue Services and GBIF UDDI registry service.

INSPIRE Metadata implementing rules differentiates between two levels of discovery metadata: level one gives general information for the non expert user, while level two is detailed enough for high level discovery by experts. The discovery purpose demands elements from the following packages,

Meta-information for the user:
Level 1:

- Citation: which resource should be located
- Identification: general description of the resource, including temporal and spatial extension properties
- Distribution: how to access the resource, and who is responsible for it
- Constraints: access and usage constraints for the dataset

Level 2:

- Dataset/service: specific description of resource (dataset or service) properties. Information about how the resource can be used
- Data quality: information about fitness for purpose of the resource
- Further meta-information to be used by services is needed to fulfil the discovery functionalities:
- Metadata: Information about the metadata
- Search: Options for querying the resource
- Navigation: Options for getting the results

- Registry transaction: Add or change registry entries
- Harvesting: Elements for automatically populating the registry or catalogue

As an example of the applicability of this model, Table 16 shows a mapping between the metadata elements currently used by GBIF (see GBIF metadata strategy_v.06.pdf154), suggested descriptors for a GBIF Profile of EML (GBIF-EML-Profile-table__v02-2.pdf) and the discovery meta-data elements.

Table 16: Mapping between metadata elements used by GBIF, suggested descriptors for a GBIF Profile of EML and discovery meta-data elements.

Current GBIF meta-data element	Suggested descriptor for GBIF Profile of EML	Proposed LifeWatch meta-data elements grouped by package	Comment
Provider binding (DiGIR, BIOCASe, TAPIR)	distribution.online. connectionDefinition	distribution. locator. protocol	
Name (of dataset)	Title	citation.title, identifica-tion.otherTitles	
Website		distribution. responsibleParty. pointOfContact	Point of contact allows to specify a website for a con-tributor
Description	Abstract	identification.abstract	
Citation		citation.identifier	
How to cite this dataset		citation	
Basis of record		dataset.topicCategory	Category of the record
Access point URL	distribu-tion.online.URL	distribution.locator.URL	
added to portal			
Information updated	pubDate	citation.dateOf("publication", “update”)	
Contacts (Name, Role, Address, Email, Tele-phone	- creator - metadataProvider - associatedParty - role	distribution.responsibleParty	For LifeWatch several con-tacts and their roles can be given, at least one contact must be specified
Data networks			
Occurrences record in-dexes (GBIF index)		citation.identifier	
Number of records shared by provider			
Occurrences with co-ordinates	coverage. geographicCoverage. coverage. temporalCoverage	identification.extent. geographicElement identification.extent. temporalElement	GBIF current model extracts the geographic reference from the protocol tags, if they are available
Occurrences with no spatial issues			
Number of species			
Number of taxa	coverage.	Could be extracted from a	Taxonomic coverage is not

[154] These PDFs can be found at the following URL: http://wiki.gbif.org/dadiwiki/wikka.php?wakka=GBIFMetadataProfile

	taxonomicCoverage	taxonomic coverage	applicable for all Feature Types in LifeWatch, but has a high relevance for the biodiversity area so it could be added as metadata elements of the dataset package
	language	dataset.language	
	keywordSet (keyword, keywordThesaurus)	identification. descriptiveKeywords	
	distribution.offline		
	intellectualRights	constraints. accessConstraints	
	researchProject - title, personell, abstract,...	annotation	Description of the research context in which the dataset was created including descriptors and documentation. Could be included as annotation
	methods - methodStrp, sampling, qualityControl		Describe methods followed in the creation of the dataset including quality control procedures. The quality control identifies a quality goals an describes steps to ensure that the dataset meet those standards

Note: Similar mappings should be done for DublinCore Elements and for the OpenSearch specification.

B.5.4.2 Purpose "Identification"

For scientific work, the system should provide a mechanism to recognise both the primary and the derived data in order to cite them in publications and protocols, to allow reproducing of experiments, and the unequivocal reference to the data producers.

Resources (data and processes) need to be identified unambiguously so that links between them can be recorded. Unique identification of resources provides a basic mechanism for the concept of bi-directional links between original and derived data, needed for provenance purposes. The identifiers must be kept resolvable if, e.g., servers change. The mechanism should be transferable outside the LifeWatch system for publication purposes (e.g. as citation). Hence, any schema should have a reasonable printed format, possibly using a well-known unambiguous URI format. The identifiers should be easy to create, to discover and to resolve for retrieve the associated data.

Relevant existing schemes to build on include:

- Handle System: http://www.handle.net/, proprietary protocol.
- Life Science Identifiers (LSID), http://lsid.sf.net/, Syntax defined by MIME types, standardised by EMBL, IBM, I3C and OMG
- Digital Object Identifiers (DOI) (based on Handle System), http://www.doi.org/.
- GBIF Resource Identification (GUID)
- Actionable Resource Tags (ART)

Identification is a general purpose applicable for all other purposes. Metadata elements from the package citation (e.g. identifier, title and version) could be applicable.

B.5.4.3 Purpose "Access, Storage and Invocation (Access)"

Meta-information concerning the location, the allowed protocols, and access constraints are needed for service invocation or binding. This meta-information is usually given by semantic descriptors in WDSL, OWL-S, or WSMO if the invocation should be performed automatically. The access and storage of data is a specialisation of service invocation, which will be separately defined by an own model due its relevance for the LifeWatch infrastructure. Examples for data access services are the ORCHESTRA Feature Access Service, the OGC Web Feature Service (WFS), or the GBIF occurrence record data service.

For LifeWatch, we recommend that services are all self-describing following the Service Model proposed by OGC, where each service implements the getCapabilities operation that provides the details of which operations and content it supports.

Relevant metadata elements for this purpose can be taken from the packages

- Dataset
- Service
- Transaction
- Service invocation
- Search
- Constraints
- Licences and fees

B.5.4.4 Purpose "Integration"

The purpose “integration” is relevant for data as well as for service composition. For LifeWatch, we will differentiate two mediation purposes: summarising of data (integration) and composition of services (orchestration). For integration, meta-information in form of rules for mapping between different representations of data and ensuing transformations, mediators for terminology, and a minimal and extensible meta-model for integrated data are needed. To target vocabulary incompatibility, semantic rules could be defined in form of format-wrappers, dictionaries can be used, or, as the ideal solution, an automatic deduction based on ontologies can be performed.

To facilitate integration, meta-models should define uncertainty levels for required information. For example, when LifeWatch uses spatial references for data integration, the location of an occurrence must be given. Since some occurrence data may lack this information, the attribute can be filled in (manually or eventually semi-automatically) by deducing an approximate spatial reference, e.g. a region or country. By filling this information, a predefined uncertainly level should be added, to refer to the method of given this information155.

The integration of data will be necessary at the following cases:

- Data aggregation: Content information will be integrated and aggregated to a hierarchical data model, to be analysed at different granularity. The use of a multidimensional model will be recommended, defining temporal, spatial and thematic dimensions as basis of common characteristics of the primary data, having well defined hierarchical levels resp. semantics. Thereby different analytical operations can be performed like the selection of aggregated information by constrain-

[155] See (Tomislav Hengl, Emiel van Loon, Henk Sierdsema, Willem Bouten (2008): Advancing Spatio-temporal Analysis of Ecological Data: Examples in R. In: Computational Science and Its Applications – ICCSA 2008, S. 692–707)

ing the levels of detail in one or more dimensions. The model could be built considering aspects of the Online-Analytical-Processing (OLAP) technology for Database Management Systems based on Warehouses, which will be more explicit treated on the engineering viewpoint.

- Data export: Data can be accessed through different protocols. It is not intended to store primary content information, which is supplied by data providers into the LifeWatch infrastructure156. Although users or services may demand to access data using specific protocols and formats. It seems more practicable not to go for a unique common LifeWatch format, but to support a restricted and extensible number of formats. LifeWatch services may extract the transferable information from a source format into the target format using meta-models.
- Orchestration: Data access services may deliver data of the same "content" on different formats to be processed by another service. An example is the application of a modelling algorithm, e.g. niche modelling, for data of different regions, being served by different providers. A meta-model for the extraction of the required information to be applied by the modelling service can be defined for each of the input source types.

Relevant metadata elements can be found in the packages:

- Citation
- Dataset
- Constraints
- Multidimensional model

B.5.4.5 Purpose "Orchestration

LifeWatch will perform service composition through workflows. Therefore well-described service interfaces, preferable enhanced with semantic information to facilitate service chaining is indispensable. To define a workflow, services will be selected for a combined invocation through the enactment machine. The resulting workflow can also be specified to constitute a new service. In order to perform workflows, composition rules and interoperability operations will be defined in the meta-information model supported by the ontologies.

LifeWatch differentiates between stateful and stateless services. This property should be visible through the general service description requested by the getCapability operation. For LifeWatch an additional entry for the operation to get the status information in the capability is recommendable.

The meta-information model for orchestration also handles the control for service chaining. LifeWatch applies for it the design patterns for service chaining defined by ISO/DIS 19119, with slightly different naming (The computational viewpoint will be more explicit on this theme).

Relevant metadata elements can be extracted from the packages:

- Citation
- Service
- Service operation
- Constraints
- Process

156 If additional information are provided by the user (e.g. annotations, etc.) this needs to be stored in the LifeWatch repository and linked to the source data by GUID.

B.5.4.6 Purpose "Personalisation"

LifeWatch is intended to serve a distributed and multidisciplinary user community. Users may be organised by virtual organisations like collaborative networks or teams working on a common e-Lab. It is important that those user groups and even individual user can define preferences for the appearance, language and functionality of the e-services, which are used. Also a widespread feature is the storage of the application or task status at the end of a session to be continued by the next log in, where a cache saves the results of the work done so far ORCHESTRA defines meta-information for user-profiling using the concept of subjects. A subject is defined as an "abstract representation of a user or software component in an ORCHESTRA application. Subject attributes are intended to store generic information about subjects (e.g. first name, last name, address, e-mail, …)". Subjects are a main concept for the purpose or authentication and authorisation, but authentication and user meta-information should be decoupled. Meta-information for subjects to be used by user profiling can be the user-groups, the used dictionaries (or ontologies), the language, information about a certain workspace and state of recent user tasks. "Apply some business logic that leverages the user's profile data and preference information along with their browsing habits to present them with the scenario that fits them"157.

Relevant metadata elements are user related, e.g. packages user profiles and user.

B.5.4.7 Purpose "Quality evaluation"

Since LifeWatch is not only a platform for data exchange but also for the use of data, information about the relevance and applicability for scientific research is a critical system capability. The EML specification defines several major sections of metadata content in order to fulfil the evaluation level.

To evaluate the quality of information and quality of services (resources in general) the following information may be relevant:

- Origin (author, identification, version)
- Citations (how the resources has been used)
- Annotations (User entries to the resources)
- Original structure
- Changes
- Validity period
- Availability (Use restrictions, access faults, mirrors)
- Uncertainty budget: Information about the inherent uncertainties (or noise) in the input data and the added uncertainty through operations like aggregation, modelling or integration
- Reference Lists

The ideal case of LifeWatch discovery results is to give users a hint (e.g. as fitness-for-purpose attribute) of the relevance of the returned entries based on the information outlined above, similar to the results of web search engines.

Relevant metadata elements can be extracted from the quality control and the annotation packages.

---

157 http://blogs.conchango.com/rizwantayabali/archive/2007/10/31/Personalisation-vs-Customisation.aspx

B.5.4.8 Purpose "Provenance"

The capability of giving provenance information relates to other purposes, but due to its outstanding relevance for LifeWatch and as "crucial component of workflow systems“158, it is treated separately.

LifeWatch defines provenance as all automatic annotated information about the usage of resources. It can be seen as the LifeWatch laboratory book. Since a resource is known to the infrastructure, the resource should be attached to a unique identification. All access and changes of the resource will be logged. Once the resource has been used, it cannot be changed or deleted. Therefore, validity time stamps may be applied and a modified copy with a reference to its parent will be created. Provenance should allow a double-reference between resources and its usages, e.g. as citations, as derived data or products on user annotations, etc.

For provenance metadata, elements from the citation and annotation packages are relevant.

B.5.4.9 Purpose "Authentication, Authorisation and Accounting (AAA)"

To restrict and control the access to resources meta-information related to authentication, authorisation and accounting mechanisms (AAA) is needed. As mentioned before this information should be separated from user data (subjects). ORCHESTRA uses the concepts of principals. "A principal is an identity of a subject whereas authentication indicates whether a subject is allowed to use a certain principal. One subject may have multiple principals. Each authentication mechanism can have its own way of representing a principal."

Authentication is the process of identifying and verifying a subject based on credentials, in the ORCHESTRA model, that means to prove, that a subject "is allowed to act with the corresponding principal" (OA-RM 7.5.3). A principal is for instance a user of an Authentication System, which can be associated to a subject. To a successfully authentication follows a session start, whereby the session information refers to the principal, which has been authenticated.

Authorisation is the process of verifying if a subject has access to a particular resource. In the sense of ORCHESTRA "a service may request an authorisation decision for a given principal and a given authorisation service context". The meta-information model can use concepts like permissions (ORCHESTRA) or licenses agreements (OGC Abstract Specification for Geo Digital Right Management, GeoDRM-Reference Model, V. 1.0). ORCHESTRA differentiates two authorisation paradigms:

- Lookup based paradigm: "Use predefined data structures to retrieve authorisation decision. The most famous representative is the role-base paradigm." (OA-RM. 7.5.4)
- Expression based paradigms: "define a framework to specify authorisation conditions. These conditions are parameterised and evaluated in order to compute authorisation decisions.(...) The most popular representatives (...) are trust management systems."(OA-RM. 7.5.4)

Accounting is "the process of gathering information about the usage of resources by subjects", e.g. duration time of usage and size of downloaded data. "Accounting information can be used to support billing, fair-use, planning and many other purposes. (...) Meta-information related to accounting is usually a combination of the principal identifying a subject (...) and some measures for resource utilisation (...) ". (RM-OA 8.4.2.7)

Relevant metadata packages are constraints and licensing.

---

158 Moreau, L., Freire, J., Futrelle, J., McGrath, R., Myers, J. and Paulson, P. (2007) The Open Provenance Model. Technical Report UNSPECIFIED, ECS, University of Southampton

B.5.4.10 Purpose "Data capture"

This purpose should support the ability of automatically extracting information from data sources like devices like sensor and mobile devices. Besides the metadata needed for access purposes notification mechanisms are needed to refer to data changes or the need for updates from the data sources. Metadata elements specifying particular datasets (e.g. for sensor data) can be relevant as well as elements from the messaging package.

Editor's Note: *The following aspects need further study:*
a) legacy data capture (e.g. on-demand digitisation of archival material, specimens, literature)?
b) on-demand molecular data capture (on-demand DNA extraction and sequencing).

B.5.4.11 Purpose "Collaboration"

The infrastructure should provide different communication mechanisms between users working on common tasks. Therefore, an information model holding the properties to be shared among collaborative tools of users and their associations to groups is needed. For collaboration meta-information elements from the packages user, user profile, messaging, and annotations may be relevant.

B.5.4.12 Purpose "Human Interaction"

LifeWatch will support the development and use of open, service oriented applications, that. In that way applications can be built by integrating different tools and by using services for communication between them. To support very thin clients (e.g. a html client) the major application logic will be provided by services. Human interaction can be measured by defining user actions, which are performed on a graphical interface of an application. Metadata can be used to specify the action (e.g. mouse-pressed or button-click), the context of the action (application menu, map application, etc.) and further user data. The result of actions is often performed as a visualisation of information on the graphic display. Metadata to support a standardised representation of the information (portrayal package) and supporting user preferences (e.g. language, disabilities support) (personalisation package) is also relevant for this purpose.

Appendix C. LifeWatch service types

C.1 An Example for a Service Type Definition: Catalogue Service

The description of the Catalogue Service is included to give an idea of the style of specification of services. The original specification can be found on the ORCHESTRA web site159.

Note: The ORCHESTRA Specification for CatalogueService can be modified by extracting the functional modules grouped into the interfaces into individual services, e.g. PublishingService, HarvestingService, and SearchService to provide more flexible architectures based on choreographies rather than on inheritance.

C.1.1 Textual Presentation

The following Table 17, copied from RM_OA, Section 9.6.6 demonstrates the style of textual, human readable presentation of a service.

Table 17: Textual high-level presentation

Name	Catalogue Service
Standard Specifications	OASIS UDDI Version 3.0.2 Specification (http://uddi.org/pubs/uddi_v3.htm) OGC 04-021-r3 Catalogue Service Implementation Specification V2.0.1 (Class: Abstract Specification) OGC 04-017r1 Catalogue Services – ebRIM (ISO/TS 15000-3) profile of CSW (CAT2 AP ebRIM) V0.9.1 (Class: Engineering Specification) OGC 04-038r2 ISO19115/ISO19119 Application Profile for CSW 2.0 (CAT2 AP ISO19115/19)) V0.9.3 (Status: Best Practices) OGC 06-079r2 EO Application Profile for CSW 2.0 (Status: Pending) OGC 06-131 EO Extension Package for ebRIM (ISO/TS 15000-3) Profile of CSW 2.0 (Status: Discussion Paper) Note: ORCHESTRA specifies a Catalogue Service that has been derived from the approach for how meta-information is being handled in the ORCHESTRA Architecture (see section 8.4 of OA-RM). Thus, the above standards have been considered, but the goal has not been to specify another variant of the OGC Catalogue Specification. The ORCHESTRA Catalogue Service does not define a meta-information schemaName Catalogue Service

Table 17 continues …

159 http://www.eu-orchestra.org/download.php?file=docs/OA-Specs/Catalogue_Service_Specification_v1.1-BRGM-IITB.pdf

Name	**Catalogue Service**
Description	The Catalogue Service supports the ability to publish, query and retrieve descriptive information (meta-information) for resources (i.e. data and services), meta-information about ORCHESTRA Source Systems (just like meta-information for other ORCHESTRA services) and instances of feature types that are referred to by extensions of the OMM_FeatureType, such as documents, schemas, dictionaries, equations and models. The Catalogue Service is not tied to a particular schema of a meta-information standard (e.g. ISO 19115); instead, it supports application schemas for meta-information (OASMI) that are designed according to the rules of the OMM. Due to independence from a specific meta-information standard, the catalogue can be used to store meta-information about services and data according to the meta-information schema used in the catalogue. Therefore, a catalogue instance can be used as a data catalogue, service registry or both if multiple meta-information types are used in the catalogue instance. The multilinguality of the catalogue is dependent on the multilingual capabilities of the meta-information schema used inside the catalogue. Meta-information entries in catalogues represent resource characteristics that can be queried and presented for evaluation and further processing by both humans and software. The Catalogue Service supports the discovery of registered re- sources within an information community and returns binding information that allows a user to locate and access the resource (e.g. an URI). The Catalogue Service provides its functionality through the following interfaces: ServiceCapabilities: Informs about the common and specific capabilities. CatalogueSearchInterface: The interface for search provides a means for searching information in the catalogue. The client asks the catalogue capabilities for the available catalogue entry types. Each entry type is associated with a meta-information type and its corresponding query languages. With this information, the client can query the catalogue entry type with the appropriate query language. CataloguePublicationInterface: The interface for publication is responsible for including, updating, and deleting meta-information in the catalogue. It is pushing information into the catalogue. It provides operations for filling the catalogue. The needed meta-information could be created with some kind of meta-information editor, in which the user is specifying the meta-information about resources to be registered in the catalogue, or it could be collected through the collection interface. CatalogueCollectionInterface: The collection interface provides operations, which are helpful for the automatic update of catalogue content in difference to the publication interface, which just fills the catalogue with given content. It is pulling meta-information into the catalogue. The operations in this interface should be able to be triggered from the outside of the catalogue and it should be possible to define a periodic update from the catalogue content. CatalogueNavigationInterface: With the means of this interface, the user is looking for meta-information records managed by the catalogue by navigating from node to node. The catalogue itself drives the search: no query is performed. Note that the implementation of this interface makes the Catalogue Service a stateful service. AsynchronousInteraction (OA Basic Service): Definition of a uniform way to request asynchronous execution of a service operation, e.g., for operations that are time-consuming or deliver results periodically. This interface is used by the collectMetaInformationPeriodic operation of the CatalogueCollectionInterface.
Interface ServiceCapabilities	
getCapabilities	Informs the requester about the common and specific capabilities of a Catalogue Service instance. Examples of specific capabilities are the information about query languages and meta-information types used in the Catalogue Service instance.

Interface CatalogueSearchInterface	
search	Returns a list of identifiers for corresponding features, given a request expressed in a given query language. Returns associated meta-information instances, given some identifiers of features managed by the catalogue as returned by a previous search operation call.
getMetaInformation	
getQueryDomain	Returns the domain of values that are applicable to a property of the meta-information type. This is used by catalogue clients. Using this operation by giving the parameters of interest, the client shall know what values (e.g. list of values, range of values) are allowed for a meta-information property.
getMetaInformationType	Returns the associated meta-information type, given a list of catalogue entry types managed by the catalogue.
Interface CataloguePublicationInterface	
createMetaInformation	Pushes information into the catalogue. The task of this operation is to insert catalogue content into the catalogue. The operation receives the meta-information to be stored and returns information about the update of the catalogue.
setMetaInformation	Updates the catalogue content. The operation receives the meta-information types to be stored and returns information about the update of the catalogue.
deleteMetaInformation	Deletes catalogue content from the catalogue. The input is a constraint to identify the catalogue content, which needs to be deleted. The operation returns information about the update of the catalogue.
Interface CatalogueCollectionInterface	
collectMetaInformation	Pulls meta-information into the catalogue. The operation receives one reference of a source of meta-information and a catalogue entry type. This catalogue entry type is the type in which the meta-information is going to be stored in the catalogue. The operation returns information about the update of the catalogue.
Information Periodic (optional)	Receives one reference of a source of meta-information, the catalogue entry type and the time interval between two collections and a date to stop the collect. The catalogue entry type is the type in which the meta-information is going to be stored into the catalogue. The operation is processed periodically according to the given intervals and stores the resulting meta-information into the catalogue. The operation should be called asynchronously using the Asynchronous Interaction interface. The operation returns information about the update of the catalogue.
Interface CatalogueNavigationInterface	
getNavigationRoots	Returns the catalogue entries that can be used to start navigation inside the catalogue. If none is returned, no navigation will be possible.
getNavigationEdges	Returns all relationships that start from this node to other ones given an existing node in the catalogue. Each relationship is annotated by the kind of relationship, which adds some semantic information (e.g. broader, narrower, similar) to the link.
Interface AsynchronousInteraction (from OA Basic Service)	
invokeAsync	Starts asynchronous execution of the collectMetaInformationPeriodic operation of the CatalogueCollectionInterface. The invokeAsync operation returns immediately with an identifier (invocation ID) representing the asynchronous execution. abort Aborts execution of the previously invoked asynchronous collectMetaInformationPeriodic operation identified by its invocation ID.
notify	Passes a notification to the callback interface provider of the CatalogueCollectionInterface.
Example usage	A possible usage scenario of the Catalogue Service is the usage of a catalogue for discovering maps and displaying them in a map viewer. The following steps need to be accomplished for this scenario:

	The catalogue needs to be initialised with meta-information about the maps and a service capable of displaying the maps. The meta-information can be written into the catalogue using operation createMetaInformation. The user performs a search for available maps on the catalogue using the search and getMetaInformation operations. The user performs a search for an available map viewer, again using the search and getMetaInformation operations. The user displays the maps in the map viewer, using the retrieved meta-information about the maps and the map viewer.
Comments	The abstract specification leaves the question of the meta-information creation open. It could be created by the user with the help of a meta-information editor or automatically either within the catalogue inside collectMetaInformation or with the usage of other means and services inside collectMetaInformation. The support of multi-linguality depends on the meta-information schema used in the catalogue. Meta-Information about data and services inside the scope of an OSN will be described with the help of the service capabilities.

C.1.2 Abstract Service Description

The abstract service description for the ORCHESTRA catalogue service is provided by the document "Specification of the Catalogue Service V. 1.1".160

The abstract service description for ORCHESTRA services should be based on the "Abstract Specification of ORCHESTRA Service Types, V. 2.2 and comprises the following structure:

- Role and scope of the service
- Context of the service (relation to standards, ongoing initiatives, and to ORCHESTRA specifications)
  Requirements
- Specification of interfaces, including the specification of the operation and parameter types
- Specification of the service specific capabilities
- Abstract Test Suite (mandatory)
- UML Models (normative)

Service Interfaces for Catalogue Service

The catalogue service is comprised of the following interfaces as shown in Figure 58:

- Service Capabilities: general service meta-information
- Catalogue Search Interface: provides means for searching information in the catalogue e.g. query language
- Catalogue Publication Interface: Responsible for including, updating and deleting meta-information in the catalogue
- Catalogue Collection Interface: provide operations for the automatic update of catalogue content
- Catalogue Navigation Interface: makes the ORCHESTRA Catalogue Service a stateful service: the meta-information refers to catalogue properties and not to the content. Navigation links are annotated with semantic information
- Catalogue semantic interface: provides semantic information in form of keywords related to the search request.

160 http://www.eu-orchestra.org/download.php?file=docs/OA-Specs/Catalogue_Service_Specification_v1.1-BRGM-IITB.pdf) [OA_Catalogue1.1

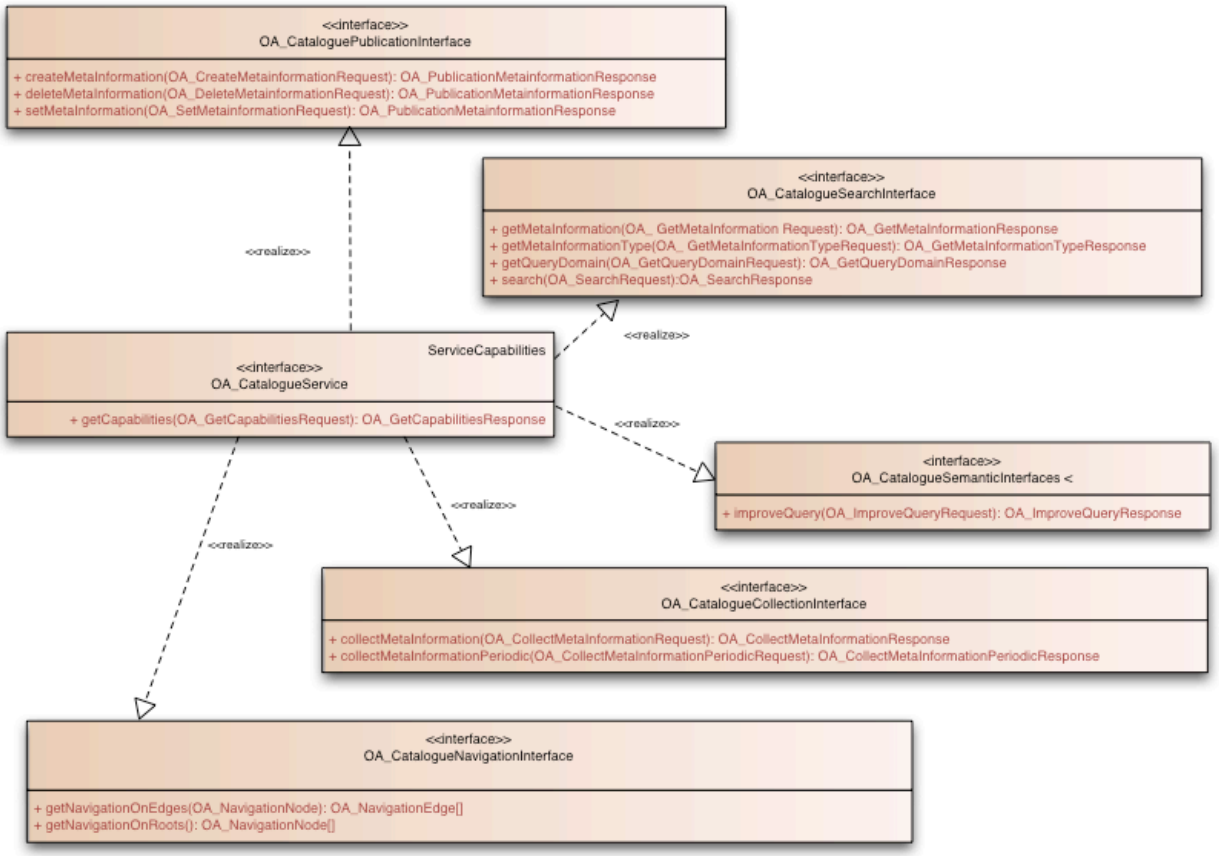


Figure 58: ORCHESTRA Catalogue Interfaces [OA_Catalogue1.1 section 3.1]

Specification of the CatalogueSearch Interface

As an example of an interface specification, the CatalogueSearch Interface will be described:

Inherited interfaces:

Interface Name	Operation Name	Specialisation
Service Capabilities	GetCapabilities	No

Operation description (see Figure 59):

- search (mandatory): given a request in a given language, returns a list of identifiers for the corresponding features
- getMetaInformation: given some feature identifiers returns the associated meta-information instances
- getQueryDomain: returns the domain values that are applicable to a query parameter (allowed values of a property of a meta-information type)
- getMetaInformationType: given a list of catalogue entry types returns the associated meta-information type

Each operation is specified according to the data types it uses, the preconditions, post conditions, usage, error codes, parameters, and return types. As an example the search operation will be specified

Specification of the Search Operation

Figure 59 shows the class diagram with the data types used by the search operation.

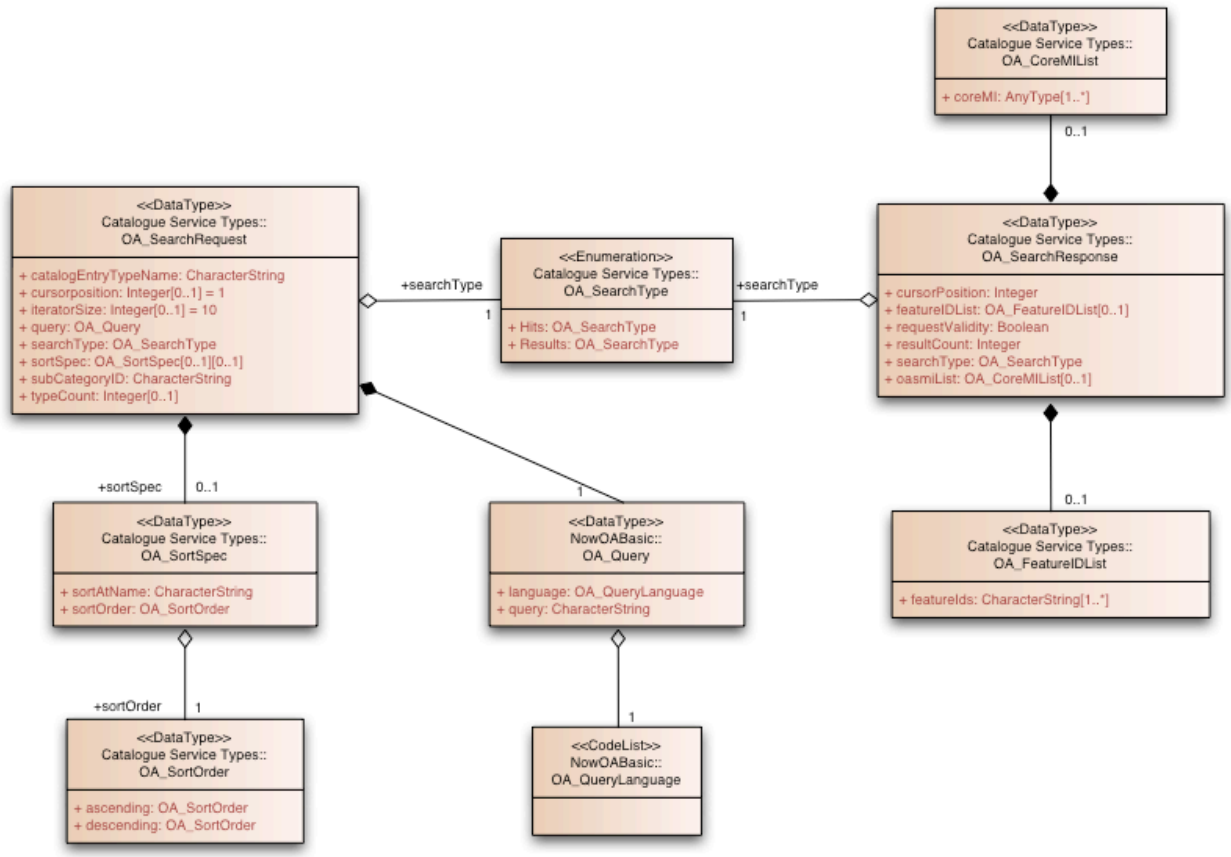


Figure 59: Class diagram for search operation [OA_Catalogue1.1 section 3.1]

Operation definition:

OA_SearchResponse search (OA_SearchRequest) throws
OA_InvalidParameterValue, OA_MissingParameterValue, OA_NotApplicableCode,
OA_InternalError, OA_UnsupportedCatalogueEntryType, OA_UnsupportedQueryLanguage
OA_UnknownSubCategoryId, OA_InvalidCursorPosition

Operation description:

Table 18: Description of the Catalogue Service Search operation

Overrides	**not applicable**
Preconditions	catalogue entry type is known from the getCapabilities operation available query languages is known from the getCapabilities operation possible values for query parameters can be obtained from the getQueryDomain operation
Postcondition	Returns either identifier of found results or just a result count
Use	Mandatory
Receives	request (OA_SearchRequest), mandatory
Returns	OA_SearchResponse
Throws	OA_InvalidParameterValue: return the name of the parameter with invalid value OA_MissingParameterValue: return the name of missing parameter OA_NotApplicableCode: thrown when no other basic or service specific exception type applies OA_InternalError: thrown when a problem occurs in the runtime environment (e.g. a out of memory) OA_UnsupportedCatalogueEntryType: The given catalogue entry type is notsupported OA_UnsupportedQueryLanguage: the query language used in the request is not supported OA_UnknownSubCategroryID: Given id is not known by the catalogue OA_InvalidCursorPosition: The cursor position given in the request is not valid regarding the current search results.

Specification of the service specific capabilities

The service capabilities reflect the meta-information schema for the service, which has a common part for all services and a specific part, specialised in each service specification. The CatalogueService service specific capabilities (OA_CatalogueCapabilities) contain information about the structure of the meta-information and the query languages supported. Figure 60 shows the class diagram for capabilities.

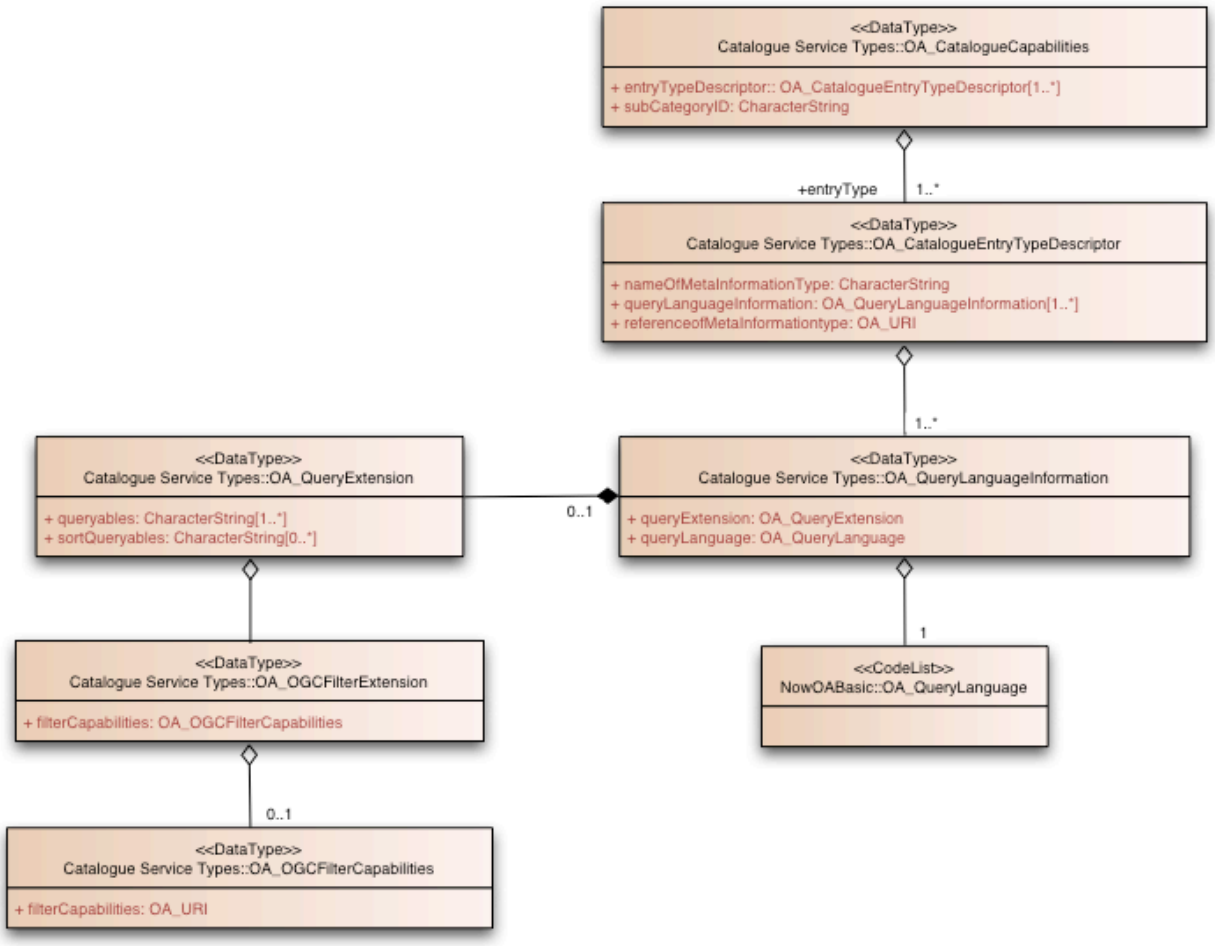


Figure 60: Class diagram for capabilities

C.1.3 Abstract test suite

The abstract test suite contains tests to ensure the conformance of the implementation to the platform-neutral abstract specification. Two example of conformance are syntactical conformance (are all mandatory operations and parameters present and are their signatures correct?) and semantic conformance (is the behaviour of the operation still the same?).

C.1.4 Service Implementation Specification

The implementation specification is based on the abstract service specification for the particular service. As an example, we will refer in this section to the ORCHESTRA implementation specification of the catalogue service for the Web Service Platforms (pilots phase 2), V. 0.8. It is based on the rules defined in the specification of the ORCHESTRA Web Service Platform and defines a concrete ORCHESTRA Meta-Information Schema (OAS-MI). The specification defines:

a. Implementation constraints: the query language defined by OGC Filter Encoding
b. Which interfaces will be implemented
c. Mapping from the abstract specification:

The mapping describes derivation from the abstract specification (operation names, parameter names, usage, cardinality) and can be implemented via XSLT-style sheets. The implementation specification can be used as a basis for different kinds of implementations (e.g. as a wrapper around an OGC Catalogue Service for the Web (CS-W) or around a Web Service Registry like UDDI). The specification provides

further XML schema documents (as profiles of GML 3.1) and a WSDL document defining the mandatory SOAP binding according to the rules of the ORCHESTRA Web Services Platform.

C.1.5 Relevant Standards for Implementation Specs of Catalogue Services

The OGC Catalogue Service for the Web (CS-W)161 is the current162 OGC implementation specification based on the abstract OGC Catalogue Specification. It establishes a general framework for implementing catalogue services. It foresees the use of different metadata information models, based for example on ISO 191xx series or on profiles of the "e-business Registry Information Model" (ebRIM)163, an OASIS information model standard (and Version 2.0 is also an ISO/TC standard –ISO/TC 15000-3) for documenting and managing metadata objects in a Web Registry.

ebRIM has been adopted by the OGC as the only cataloguing meta model for OGC CS-W catalogue implementations. The CSW-ebRIM Service Registry Service profile is the OGC implementation specification standard, extending the OASIS ebRIM Registry Information Model V.3.0 to be used by CS-W. Other models specifications based on the ebRIM ISO/TS 15000-3 profile, the FGDC CSDGM Application profile and the ISO19115/ISO19119 application profiles have been set as deprecated by the OGC.

CS-W is based on a distributed query architecture, which can lead to performance limitations, since the search is at best limited to the slowest server and to a least denominator of implementations164. That is because in a distributed query each server needs to implement exactly the same (OGC filter) functionality.

There are also catalogue service implementations based on the Web Feature Service (WFS), since metadata can be encoded as features using the General Feature Model (ISO 19101). The implementations based on WFS have often a better performance but lack more catalogue and metadata specific functions.

The 'Open Archives Initiative Protocol for Metadata Harvesting' (OAI-PMH) is a simple and strictly RESTful harvesting protocol based on Dublin Core initiated by libraries, universities, museums and galleries to 'open access' (OA) free online availability of digital content. It does not provide a search protocol, thus it can be combined with WFS-based catalogue implementation enhancing the catalogue functionalities. Other combinations could be based on OpenSearch/GeoRSS or SRU/SRW. SRU/SRW are twin protocols replacing a very widely used client-server based protocol, the ANSI/NISO standard Z39.50 based on the HTTP protocol.

Web publishing technologies like GeoRSS165 feeds and the Atom Publishing Protocol (APP)166 has been used as output formats for by best practice implementations of OGC Web services (for example the Canadian geospatial data infrastructure and the geoss esri geoportal prototype).

---

161 http://www.opengeospatial.org/standards/cat

162 The current version is the OGC Catalogue Service 2.0.2 specification (OGC 07-006r1)

163 http://www.oasis-open.org/specs/index.php#ebxmlrimv3.0

164 Source http://gis.hsr.ch/wiki/OAI-PMH

165

166 http://tools.ietf.org/html/rfc5023

Blank Page

C.2 Comparative classification of LifeWatch services

C.2.1 Taxonomic Classification of Services according to ISO 19119

Services can be grouped into categories according to the semantic type of computation they provide; in other words, by the collection of interfaces they support. The service taxonomy can be used to facilitate discovery and workflow modelling, since it describes common functional properties through interfaces. It also facilitates decisions about whether services can be substituted for other services. ISO 19119 defines taxonomy of six classes of ICT services to categorise geographic services [OGC 02-112]:

- Human interaction services are services for management of user interfaces, graphics, multimedia, and for presentation of compound documents (e.g. viewers and editors);
- Model/Information management services are services for management of the development, manipulation, and storage of metadata, conceptual schemas, and data sets;
- Workflow/Task services are services for support of specific tasks or work-related activities conducted by humans. These services support use of resources and development of workflows involving a sequence of activities or steps (chain definition and enactment);
- Processing services are services that perform large-scale computations involving substantial amounts of data. Examples include services for providing the time of day, spelling checkers, and services that perform coordinate transformations (e.g., that accept a set of coordinates expressed using one reference system and converting them to a set of coordinates in a different reference system). A processing service does not include capabilities for providing persistent storage of data or transfer of data over networks. The geographic service taxonomy differentiates four subclasses of processing services:
    - Spatial processing services;
    - Thematic processing services;
    - Temporal processing services; and,
    - Metadata processing services.
- Communication services are services for encoding and transfer of data across communications networks;
- System management services are services for the management of system components, applications, and networks. These services also include management of user accounts and user access privileges

C.2.2 Grouping services according to ISO 19119

Existing abstract as well as implementation specifications for general services will be used whenever possible in order to minimise the effort of the developing of new LifeWatch specifications. All the specified ORCHESTRA Basic Services will be considered as LifeWatch Basic Services. Additional Basic Services will be derived from the biodiversity domain and added to this set.

The OGC Reference Model (OGC 03-040 ORM_version 0.1.3) and the INSPIRE Network service Architecture (DT Network Services V 3.0) introduce service categories to be applied for spatial related services. A refinement of the ISO service taxonomy by building a category hierarchy is helpful, since services can be described more precisely (e.g. through common interfaces) and the hierarchy can be used by the meta-information model, for mediation, and for defining rules for the specification of service types in the meta- model. LifeWatch uses the purposes listed in the information viewpoint as a basis for the definition of the sub-categories.

Table 19 below shows mappings between the ISO 19119 service taxonomy and existing or planed. The first column refers and describes to the ISO taxonomy. The second column lists an extract of specified OGC167 services types, grouped into sub-categories (written in bold). The third column lists ORCHESTRA service types grouped by the ORCHESTRA service category168 (and written in bold). The forth column lists existing (and planed) biodiversity service instances. The right-most column, lists LifeWatch Basic Services and LifeWatch service taxonomic sub-categories written in bold. The elements in this column are by no means exhaustive, since the definition of the LifeWatch Basic Service will be extended during the preparatory phase and corroborated during the construction phase. It shows the subcategories in relationship with the ISO 19119 service taxonomy, the categories defined by the INSPIRE directive with the corresponding ORCHESTRA service types as well as existing or planned services from the biodiversity community.

Table 19: Relations between service categories in different domains

OGC Sub-Categories and Services Types	**ORCHESTRA Service Categories and Services Types**	**Biodiversity Services Instances**	**LifeWatch Sub-Categories and Services Types**
ISO 19119 Service Taxonomy - Human interaction services: Management of user interfaces, graphics, multimedia, and presentation of compound documents			
Application services • Discovery application services • Map Viewer Application services • Value-add application service • Imagery exploitation application service • Chain definition editor • Workflow enactment manager • Mobile location services Portrayal services • Map portrayal services • Coverage portrayal services	OA-Info Structure Service • Map and diagram service	Mapping Services EDIT MapRestService	Portrayal services • Map service • Reporting service Interaction services • Thematic application service (e.g. mapping applications, thesaurus, modelling applications) Personalisation services • Language support service • Colour schema service Collaboration services • Mailing service • Calendar service • Project management service
ISO 19119 Service Taxonomy - Model/Information management services: Management of development, manipulation and storage of metadata, conceptual schemas, and datasets			
Data services • Feature access service • Map access service • Coverage access service • Sensor description service • Sensor collection service • Product access service	OA-Info-Structure • Feature access service • Document Access Service • Sensor access service • Catalogue service • Name service OA-Support • Ontology access service	Vocabulary Data Access • GBIF taxon data service • BODC NERC datagrid vocabulary service • WoRMS web service for marine taxonomy Biodiversita Data Access • GBIF occurrence data service	Data Access Service • Document Access Service • Taxonomy Access Service • Feature Access Service • Thematic Data Access Service (e.g. for sensor data, species occurrences, genomic data, dictionaries,

[167] See http://www.opengeospatial.org/

[168] Orchestra Architecture (OA) Info-Structure and Support services. For definition see Reference Model V2 Rev 2.1, Section 9.3

OGC Sub-Categories and Services Types	ORCHESTRA Service Categories and Services Types	Biodiversity Services Instances	LifeWatch Sub-Categories and Services Types
• Feature type service • Gazetteer service • Order handling service • Standing order service • Image archive service Registry services • Catalogue service • Registry service	• Thesaurus access service • User management service • Annotation service	• CDM RestServices • EDIT Common Data Model • Services for TDWG Ontology • GEOSS web services • GMES Marine Core services Checklist access • CoL Annual Checklist Web Service • CoL Dynamice Checklist Web Service Catalogue Services • Common Service Catalogue	etc.) Annotation services • User annotation service • Provenance service Identification services • Naming service Discovery Services • Catalogue Service Mediation Services • Ontology Access Service User Management Services • User Management Service
ISO 19119 Service Taxonomy - Workflow/Task service: Support specific tasks or work-related activities			
Processing services • Chain definition services • Workflow enactment services • Subscriptions services	OA-Support • Service chain access service		Orchestration services • Workflow definition service • Workflow enactment service
ISO 19119 Service Taxonomy - Processing services: Performance of computations.			
Processing service • Coordinate transformation services • Geocoder services • Gazetteer services • Geoparser services • Route determination services • Sampling service • Thematic classification services • Feature generalisation services • Spatial counting service • Seographic information extraction services • Temporal proximity services • Geographic annotations • Statistical calculation	OA-Support • Coordinate Operation Service • Gazetteer Service • Schema Mapping Service	• GBIF Occurrence density data service • GBIF Dataset metadata service • World Gazetteer Service • EDIT occurrence to distribution converter • EDIT sorting service for taxa lists • BODC Marsden Square translator service • Open Modeller Web services	Spatial processing services • Gazetteer service • Coordinate transformation service • Geoparser service • Geocoder service • Geolinking service • Coverage generalisation • Feature generalisation service Temporal processing services Taxonomic processing services Thematic processing services • Modelling services • Interpolation service • Image classification service • Occurrence distribution map service • Support pressures and drivers analysis

OGC Sub-Categories and Services Types	ORCHESTRA Service Categories and Services Types	Biodiversity Services Instances	LifeWatch Sub-Categories and Services Types
			Metadata services • Metadata extraction service Integration services • Cube definition service • Thematic generalisation service • Statistics service Taxonomic processing services • Ecological processing services
ISO 19119 Service Taxonomy - Communication services: Encoding and transfer of data across networks			
• Encoding services • Transfer services • Geographic compression services • Geographic format conversion services • Web Notification service	OA-Support • Format Conversion Service	•EDIT CDMExportService	Encoding services Web encoding service Compression service Transformation services • Format Transformation service Messaging services • Notification Service Transfer services
ISO 19119 Service Taxonomy - System management services: Management of system components, applications and networks (including access control)			
	OA-Info-Structure • Service Monitoring Service • Authentication Service • Authorisation Service		Monitoring Services • Logging service Service Management Services • Execution Management Serv. • Resource Management serv. Transaction Services • Transaction service Quality Evaluation Services • Quality service Security Services • Authorisation Service • Authentication Service

Appendix D. Acronyms and abbreviations

The following acronyms and abbreviations are used throughout the present document.

AAA	Authentication, Authorisation, and Accounting
ABCD	Access to Biological Collections Data (TDWG standard)
ACID	Atomicity, Consistency, Isolation, and Durability
ADC	Architecture and Data Committee (GEOSS)
ALTERNet	A Long-Term Biodiversity, Ecosystem and Awareness Research Network (EU Project)
API	Application Programming Interface
BioCASE	A biological collection access service for Europe (EU Project)
CDM	Common Data Model
CEN	Comité Européen de Normalisation (European Committee for Standardization)
CORBA	Common Object Request Broker Architecture
CSL	Conceptual Schema Language
DCP	Distributed Computing Platform
DiGIR	Distributed Generic Information Retrieval
DIS	Draft International Standard
DoW	ORCHESTRA Description of Work
DRM	Digital Rights Management
DwC	Darwin Core (TDWG standard)
EBAC	Expression-based access control
EC	European Commission
EGEE	The Enabling Grids for E-sciencE
EML	Ecological Metadata Language
ESA	European Space Agency
ESDI	European Spatial Data Infrastructure
GBIF	Global Biodiversity Information Facility
GeoDRM	Digital Rights Management related to Geographic Information
GEOSSGlobal	Earth Observation System of Systems
GFM	General Feature Model
GMES	Global Monitoring for Environment and Security
GML	Geography Markup Language
GUID	Globally Unique Identifier
HCI	Human-Computer Interaction
ID	Identifier
IETF	Internet Engineering Task Force
INSPIRE	Infrastructure for Spatial Information in Europe
IOBIS	Ocean Biogeographic Information System
IS	International Standard
ISO	International Standardization Organisation
IST	Information Society Technology

IUBS	International Union of Biological Sciences
KML	Keyhole Markup Language
LMO	Legally Mandated Organisations
LSID	Life Science Identifier
LTER	Long Term Ecological Research Network
OA	ORCHESTRA Architecture
OAA	ORCHESTRA Application Architecture
OAS	ORCHESTRA Application Schema
OASIS	1. IST FP-6 project: Open Advanced System for Improved Crisis Management 2. Organization for the Advancement of Structured Information Standards
OASIS-SOA-RA	OASIS Reference Architecture for Service Oriented Architecture
OASIS-SOA-RM	OASIS Reference Model for Service Oriented Architecture
OAS-MI	ORCHESTRA Application Schema for Meta-information
ODP	Open Distributed Processing
OFS	ORCHESTRA Feature Set
OGC	Open Geospatial Consortium
OIS	ORCHESTRA Implementation Specification
OMG	Object Management Group
OMM	ORCHESTRA Meta-model
ORCHESTRA	Open Architecture and Spatial Data Infrastructure for Risk Management
OSC	ORCHESTRA Service Component
OSI	ORCHESTRA Service Instance
OSN	ORCHESTRA Service Network
OT Service	ORCHESTRA Thematic Service
OWL	Web Ontology Language
OWL-SL	Web service ontology based on OW
RBAC	Role-Based Access Control
RDF	Resource Description Framework
RM	Risk Management
RM-OA	Reference Model for the ORCHESTRA Architecture
RM-ODP	Reference Model for Open Distributed Processing
SAWSDL	Semantic Annotations for WSDL
SDD	Structured Descriptive Data (TDWG standard)
SDI	Spatial Data Infrastructure
SDIC	Spatial Data Interest Communities
SEIS	Shared Environmental Information System
SERONTO	Socio-Ecological Research and Observation Ontology
SOA	Service-oriented Architecture
SOAP	Simple Object Access Protocol
SSOA	Semantic Service Oriented Architecture
SWRL	Semantic Web Rule Language

TAPIR	TDWG Access Protocol for Information Retrieval
TCS	Taxonomic Concept Schema (TDWG standard)
TDWG	Biodiversity Information Standards (TDWG) (formerly the IUBS Taxonomic Databases Working Group)
UAA	User Management, Authentication and Authorisation
UDDI	Universal Description, Discovery and Integration
UML	Unified Modelling Language
URI	Uniform Resource Identifier
URN	Uniform Resource Name
W3C	World Wide Web Consortium
WADL	Web Application Description Language
WIN	Wide Information Network for Risk Management
WNS	Web Notification Service
WSDL	Web Service Definition Language
WSMO	Web Service Modelling Ontology
XML	Extensible Markup Language
XSD	XML Schema Definition

Blank Page

Appendix E. Glossary of terms and definitions

Editor's Note: This glossary needs to be cleaned up to remove terms that are not used in the present document.

In the following terms and definitions, the source of the term, where relevant, is given in square brackets at the end of the definition.

Access control	Ability to enforce a policy that identifies permissible actions on a particular resource by a particular subject.
	ii) An information architecture that comprises, in terms of software design, a reusable software template, or skeleton, from which key enabling and supporting services can be selected, configured and integrated with application code. [http://www.opengeospatial.org/resources/?page=glossary]
	ii) A catalogue of toponyms (place names) assigned with geographic references. A gazetteer service retrieves the geometry for one or more features, given their associated well-known feature identifiers (text strings). [http://www.opengeospatial.org/resources/?page=glossary]
	ii) Viewpoint of the ORCHESTRA Reference Model that specifies the modelling approach of all categories of information the ORCHESTRA Architecture deals with, including their thematic, spatial, temporal characteristics as well as their meta-information. [OGC 07-097; RM-OA 2007]
(Service) Platform	Set of infrastructural means and rules that describe how to specify service interfaces and related information and how to invoke services in a distributed system. [OGC 07-097; RM-OA 2007] NOTE: Examples of service platforms are Web Services according to the W3C specifications, including a GML profile for the representation of geographic information, or a CORBA-based infrastructure with a UML profile according to the OMG specifications.
Accounting	Process of gathering information about the usage of resources by subjects. [OGC 07-097, RM_OA 2007]
Application	Use of capabilities, including hardware, software and data, provided by an information system specific to the satisfaction of a set of user requirements in a given application domain. [Derived from http://www.opengeospatial.org/resources/?page=glossary]
Application Architecture	Instantiation of a generic and open architecture (e.g. the ORCHESTRA Architecture) by inclusion of those thematic aspects that fulfil the purpose and objectives of a given application. The concepts for such an application stem from a particular application domain. [From OGC 07-097, RM_OA 2007]
Application Domain	Integrated set of problems, terms, information, and tasks of a specific thematic domain that an application (e.g. an information system or a set of information systems) has to cope with. [OGC 07-097, RM_OA 2007] NOTE: One example of an application domain is management of biodiversity resources.
Application Ontology	[To be defined]
Application Schema	Conceptual schema for data required by one or more applications. [ISO/FDIS 19109:2003]
Architecture (of a system)	Set of rules to define the structure of a system and the interrelationships between its parts. [ISO/IEC 10746-2:1996]
Architecture Service	Service that provides a generic, platform-neutral and application-domain

	independent functionality. [Derived from OGC 07-097; RM-OA 2007]
Authentication	Concerns the identity of the participants in an exchange. Authentication refers to the means by which one participant can be assured of the identity of other participants. [SOA-RA, 2008]
Authorisation	Concerns the legitimacy of the interaction. Authorisation refers to the means by which an owner of a resource may be assured that the information and actions that are exchanged are either explicitly or implicitly approved. [SOA-RA, 2008]
BioCASe protocol	Distributed information retrieval protocol developed by BioCASE
Capability	A real world effect that a service provider is able to provide to a service consumer. [SOA-RM, 2006]
Catalogue	Collection of entries, each of which describes and points to a feature collection. Catalogues include indexed listings of feature collections, their content, their coverage, and of meta-information. A catalogue registers the existence, location, and description of feature collections held by an Information Community. Catalogues provide the capability to add and delete entries. A minimum Catalogue will include the name for the feature collection and the location handle that specifies where these data may be found. Each catalogue is unique to its Information Community. [Derived from http://www.opengeospatial.org/resources/?page=glossary]
Choreography	(Of services) [To be defined] cf. Orchestration
Class	Description of a set of objects that share the same attributes, operations, methods, relationships, and semantics. [ISO/IEC 19501]
Codelist	Value domain including a code for each permissible value [ISO 19136]
Common Interface	[To be defined]
Component	Hardware component (device) or Software Component.[OGC 07-097; RM-OA 2007]
Computational viewpoint	Viewpoint of an ODP system and its environment that enables distribution through functional decomposition of the system into objects that interact at interfaces. [ISO/IEC 10746-2]
Conceptual model	Model that defines concepts of a universe of discourse; an abstract description of the real world. [ISO 19109:2005(E); ISO 19101]
Conceptual schema	Formal description of a conceptual model. [ISO 19109:2005(E); ISO 19101]
Conceptual Schema Language	Formal language based on a conceptual formalism for the purpose of representing conceptual schemas. [ISO 19101]
Core Ontology	Conceptual model, covering the most important common concepts of a community that can be extent by domain ontologies.
Coverage	Function from a spatial, temporal, or spatio-temporal domain to an attribute range. Coverage associates a position within its domain to a record of values of defined data types. Thus, coverage is a feature with multiple values for each attribute type, where each direct position within the geometric representation of the feature has a single value for each attribute type. [ISO 19123]
Data	[To be defined] (See 5.3)
Data Harmonisation	Providing access to spatial data through network services in a representation that allows for combining it with other harmonised data in a coherent way by using a common set of data product specifications. [INSPIRE Directive]
Data Harmonisation Components	Structured collection of components that will be documented to support the interoperability and harmonisation of spatial data across Europe. [INSPIRE

	Directive]
Data Integration	Data integration is the process of combining data residing at different sources and providing the user with a unified view of these data.
Data Warehouse	[To be defined]
Dataset	An identifiable collection of data [ISO 19115]
Discovery	Act of locating a machine processable description of a resource that may have been previously unknown and that meets certain functional criteria. It involves matching a set of functional, semantic and other criteria with a set of resource descriptions. [Derived from W3C: http://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/#discovery]
Distributed Computer Platform	[To be defined]
Domain Ontology	[To be defined]
Drill Into	[To be defined]
Element (UML)	[To be defined]
Engineering viewpoint	Viewpoint of an ODP system and its environment that focuses on the mechanisms and functions required to support distributed interaction between objects in the system.[ISO/IEC 10746-2]
Enterprise viewpoint	Viewpoint of an ODP system and its environment that focuses on the purpose, scope, and policies for that system. [ISO/IEC 10746-2]
Execution context	The set of technical and business elements that form a path between those with needs and those with capabilities and that permit service providers and consumers to interact. [SOA-RM, 2006]
External Source System	Source system that does not provide its data and functions through an ORCHESTRA-conformant interface.
Feature	Abstraction of a real world phenomenon [ISO 19101] perceived in the context of an ORCHESTRA Application. [OGC 07-097; RM-OA 2007; derived from ISO 19101] NOTE 1: The ORCHESTRA understanding of a “real world” explicitly comprises hypothetical worlds. NOTE 2: Features may but need not contain geospatial properties. In this general sense, a feature corresponds to an “object” in analysis and design models.
Feature Catalogue	Catalogue(s) containing definitions and descriptions of the spatial object types, their attributes and associated components occurring in one or more spatial data sets, together with any operations that may be applied. [INSPIRE, ISO 19110 – modified]
Feature Service Type	[To be defined]
Feature Type	[To be defined]
Framework	A set of assumptions, concepts, values, and practices that constitute a way of viewing the current environment. [SOA-RM, 2006]
Gazetteer	Directory of instances of a class or classes of features containing some information regarding position. [ISO 19112]
Generic	(Service, Infrastructure, etc.) Independent on the organisation structure and application domain, etc. For example, a service is generic, if it is independent of the application domain. A service infrastructure is generic, if it is independent of the application domain and if it can adapt to different organisational structures at different sites, without programming (ideally). [derived from OGC 07-097; RM-OA 2007]

Geospatial	Referring to a location relative to the Earth's surface. “Geospatial” is more precise in many geographic information system contexts than "geographic," because geospatial information is often used in ways that do not involve a graphic representation, or map, of the information. [http://www.opengeospatial.org/resources/?page=glossary]
Implementation	Software package that conforms to a standard or specification. A specific instance of a more generally defined system. [http://www.opengeospatial.org/resources/?page=glossary]
Individual	[To be defined]
Information	[To be defined]
Information Community	A collection of people (a government agency or group of agencies, a profession, a group of researchers in the same discipline, corporate partners cooperating on a project, etc.) who, at least part of the time, share a common digital geographic information language and common spatial feature definitions [http://www.opengeospatial.org/resources/?page=glossary]
Information Model	The characterisation of the information that is associated with the use of a service. [SOA-RM, 2006]
Information Viewpoint	Viewpoint of an ODP system and its environment that focuses on the semantics of information and information processing. [ISO/IEC 10746-2]
Inheritance	[To be defined]
Instance	[To be defined]
Integrity	Concerns the protection of information that is exchanged – either from unauthorised writing or inadvertent corruption. Integrity refers to the assurance that information that has been exchanged has not been altered. [SOA-RA, 2008]
Interface	Named set of operations that characterise the behaviour of an entity. [ISO 19119:2005; http://www.opengis.org/docs/02-112.pdf] NOTE: The aggregation of operations in an interface, and the definition of interface, shall be for the purpose of software reusability. The specification of an interface shall include a static portion that includes definition of the operations. The specification of an interface shall include a dynamic portion that includes any restrictions of the order of invoking the operations.
Interoperability	i) Capability to communicate, execute programs, or transfer data among various functional units in a manner that requires the user to have little or no knowledge of the unique characteristics of those units [ISO 2382-1, ISO 19119:2005]. ii) Possibility for spatial data sets to be combined, and for services to interact, without repetitive manual intervention, in such a way that the result is coherent and the added value of the data sets and services is enhanced [INSPIRE Directive]
Map service	[To be defined]
Mediation	[To be defined]
Meta-information	Descriptive information about resources in the universe of discourse. Its structure is given by a meta-information model depending on a particular purpose. [OGC 07-097; RM-OA 2007] NOTE: A resource by itself does not necessarily need meta-information. The need for meta-information arises from additional tasks or a particular purpose (like catalogue organisation), where many different resources (services and data objects) must be handled by common methods and therefore have

	to have/get common attributes and descriptions (like a location or the classification of a book in a library).
Meta-information model	Implementation of a conceptual model for meta-information. It is represented by an ORCHESTRA Application Schema for Meta-information. [OGC 07-097; RM-OA 2007]
Middleware	Software in a distributed computing environment that mediates between clients and servers. [http://www.opengeospatial.org/resources/?page=glossary]
Observation	Act of observing a property or phenomenon, with the goal of producing an estimate of the value of the property. [OGC 07-022]
Observed Property	Identifier or description of the phenomenon for which the observation result provides an estimate of its value. [Derived from OGC 07-022r1]
Ontology	Explicit, formal specification of a shared conceptualisation (Studer et al 1998). It is formal in order not only to make it readable by humans, but also by machines. It is explicit as it is based on a taxonomy specified in terms of concepts, properties (or relationships), and axioms (the "vocabulary"). It is shared in the sense that these specifications are fixed as an agreement set up and shared by a dedicated user community and that it is associated with a particular subject area (domain) or task. It is a conceptualisation as it defines a conceptual schema by abstracting from a real or hypothetical world. Its ultimate purpose is to enable machine understanding, which in turn provides the potential for data and service interoperability. [Based on (Studer et al 1998)]
Open Architecture	Architecture whose specifications are published and made freely available to interested vendors and users with a view of widespread adoption of the architecture. An open architecture makes use of existing standards where appropriate and possible and otherwise contributes to the evolution of relevant new standards. [OGC 07-097; RM-OA 2007]
Open Geospatial Consortium (OGC)	A non-profit, international, voluntary consensus standards organization for the development of standards for geospatial and location based services.
Operation	Specification of a transformation or query that an object may be called to execute. An operation has a name and a list of parameters. [ISO 19119:2005; http://www.opengis.org/docs/02-112.pdf]
ORCHESTRA Application	Set of software components that together comprise an application based on the usage of ORCHESTRA Services. [OGC 07-097; RM-OA 2007]
ORCHESTRA Application Architecture (OAA)	Instantiation of the ORCHESTRA Architecture by inclusion of those thematic aspects that fulfil the purpose and objectives of a given application. The concepts for such an application stem from a particular application domain (e.g. a biodiversity research application). [Based on OGC 07-097; RM-OA 2007]
ORCHESTRA Application Implementation Specification (OAIS)	Extension and restriction of an ORCHESTRA Implementation Specification according to the needs of a particular application domain. An OAIS comprises a platform-specific combined specification of a thematic information model and a set of OT Services. [OGC 07-097; RM-OA 2007]
ORCHESTRA Application Schema (OAS)	Conceptual schema for the data required by one or more ORCHESTRA Applications. As such, it provides a formal specification that is compliant to the ORCHESTRA Meta-model of the concepts (e.g. feature types), their properties and associations which are relevant for a specific information model in an ORCHESTRA Service Network. [OGC 07-097; RM-OA 2007;extending ISO/FDIS 19109:2003]
ORCHESTRA Application	Form of an ORCHESTRA Application Schema applied to meta-information.

Schema for Meta-information (OAS-MI)	[OGC 07-097; RM-OA 2007]
ORCHESTRA Architecture (OA)	Open architecture that comprises the combined generic and platform-neutral specification of the information and service viewpoint as part of the ORCHESTRA Reference Model. [OGC 07-097; RM-OA 2007]
ORCHESTRA Architecture Service (OA Service)	ORCHESTRA Service that provides a generic, platform-neutral and application-domain independent functionality. [OGC 07-097; RM-OA 2007]
ORCHESTRA Feature Set (OFS)	Set of feature instances following the information model formally specified in an ORCHESTRA Application Schema. [OGC 07-097; RM-OA 2007]
ORCHESTRA Implementation Specification	Combined platform-specific specification of the engineering and technology viewpoints as a result of the mapping of the ORCHESTRA Architecture to a specific platform. [OGC 07-097; RM-OA 2007]
ORCHESTRA Meta-Model (OMM)	Framework of rules for the specification of an ORCHESTRA Application Schema. It is specified in terms of UML classes stereotyped as <<MetaClass> and associated rules for their instantiation in an ORCHESTRA Application Schema [OGC 07-097; RM-OA 2007]
ORCHESTRA Protocol	[To be defined]
ORCHESTRA Reference Model	The ORCHESTRA Reference Model comprises a specification of all RM-ODP viewpoints for the open architecture for environmental risk management. In particular, it encompasses the specification of the ORCHESTRA architecture and a specification framework for ORCHESTRA Implementation Specifications that are implemented by ORCHESTRA Service Components and deployed in an ORCHESTRA Service Network as ORCHESTRA Service Instances. [OGC 07-097; RM-OA 2007; http://www.eu-orchestra.org]
ORCHESTRA Service	Service specified as an ORCHESTRA Service Type, implemented as ORCHESTRA Service Component, and offered in an ORCHESTRA Service Network by an ORCHESTRA Service Instance. [OGC 07-097; RM-OA 2007]
ORCHESTRA Service Component	Component that provides an external interface of an ORCHESTRA Service according to an ORCHESTRA Implementation Specification. [OGC 07-097; RM-OA 2007]
ORCHESTRA Service Instance	Executing manifestation of an ORCHESTRA Service Component. [OGC 07-097; RM-OA 2007]
ORCHESTRA Service Network	Set of networked hardware components and ORCHESTRA Service Instances that interact in order to serve the objectives of ORCHESTRA Applications. The basic unit within an OSN for the provision of functions are the OSIs. [OGC 07-097; RM-OA 2007]
ORCHESTRA Service Type	Type of an ORCHESTRA Service specified according to the rules of the ORCHESTRA Reference Model. ORCHESTRA Service Types are functionally classified in ORCHESTRA Architecture Services (OA Services) and ORCHESTRA Thematic Services (OT Services). [OGC 07-097; RM-OA 2007]
ORCHESTRA Source System	Source system that provides its data and functions through an ORCHESTRA-conformant interface. Each ORCHESTRA Source System is associated with at least one External Source System. [OGC 07-097; RM-OA 2007]
ORCHESTRA Thematic Service(OT Service)	ORCHESTRA Service that provides an application domain-specific functionality built on top and by usage of OA Services and/or other OT Services. [OGC 07-097; RM-OA 2007] NOTE: An OT Service may but need not be specified in a platform-neutral

	way.
Orchestration	(of services) [To be defined] cf. Choreography
Policy	Representation of a constraint or condition on the use, deployment or description of a resource. [Derived from SOA-RM, 2006]
Principal	A principal represents the identity of a subject in an ORCHESTRA Service Network. A subject may have several identities, and thus several principals. The association between a principal and a subject is established in an authentication process. [Derived from OGC 07-097; RM-OA 2007]
Protocol	[To be defined]
Purpose	(of meta-information)A purpose of meta-information describes the goal of the usage of the resources. [OGC 07-097; RM-OA 2007]
R	A language and environment for statistical computing and graphics. [http://www.r-project.org/]
Reference Architecture	Reference architecture is an architectural design pattern that indicates how an abstract set of mechanisms and relationships realizes a predetermined set of requirements. [SOA-RM, 2006]
Reference Model	A reference model is a framework for understanding significant relationships among the entities of some environment, and for the development of consistent standards or specifications supporting that environment. A reference model is based on a small number of unifying concepts and may be used as a basis for education and explaining standards to a non-specialist. [http://ssdoo.gsfc.nasa.gov/nost/isoas/us04/defn.html]
Representation	Comprises any useful information about the current state of a resource. [Richardson/Ruby 2007]
Research Site	A place (i.e., a physical institution) where research tasks are carried out.
Resource	A source of capability and capacity for performing some task or providing some information that is important enough to be identified and referenced as a discrete entity, and which can be called upon to provide said capability and capacity in order to satisfy the need of another.
Resource coupling	[To be defined]
Semantic Data Integration	[To be defined]
Semantic GRID	[To be defined]
Semantic Interoperability	Semantic interoperability emphasises the importance of information inside enterprise networks and focuses on enabling content, data, and information to interoperate with software systems outside of their origin. Information's meaning is the crucial enabler that allows software to interpret the appropriate context, structure, and format in which the information should reside at any given moment and inside any given system. (Pollock, Hodgson 2004)
Semantic Service Oriented Architecture	[To be defined]
Semantic Web	The Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries. It is a collaborative effort led by W3C with participation from a large number of researchers and industrial partners. It is based on the Resource Description Framework (RDF), which integrates a variety of applications using XML for syntax and URIs for naming. [W3C; http://www.w3.org/2001/sw/Overview.html]
Semantic Web Rule Language	[To be defined] (http://www.w3.org/Submission/SWRL/)
Semantics	A conceptualisation of the implied meaning of information that requires

	words and/or symbols within a usage context. [SOA-RM, 2006]
Sensor	[To be defined]
Service	Distinct part of the functionality that is provided by an entity through interfaces. [ISO 19119:2005; ISO/IEC TR 14252; http://www.opengis.org/docs/02-112.pdf]
Service chain	Sequence of services where, for each adjacent pair of services, occurrence of the first action is necessary for the occurrence of the second action. [ISO/DIS 19119
Service Component	Software component that provides an external interface of a service according to an implementation specification for a given platform. [Derived from OGC 07-097; RM-OA 2007]
Service Instance	Executing manifestation of a software component that provides an external interface of a service according to an implementation specification for a given platform. [OGC 07-097; RM-OA 2007]
ServiceMapping	Process of mapping a description of a Service Type and the specification of its interfaces on platform-neutral level to an Implementation Specification for a given platform [derived from OGC 07-097; RM-OA 2007]
Service Ontology	[To be defined]
Service Oriented Architecture (SOA)	Service Oriented Architecture is a paradigm for organising and utilising distributed capabilities that may be under the control of different ownership domains. It provides a uniform means to offer, discover, interact with, and use capabilities to produce desired effects consistent with measurable preconditions and expectations. [SOA-RM, 2006]
Service Viewpoint	Viewpoint of a Reference Model that specifies services supporting the syntactic and semantic interoperability between source systems and the development of an application. [Derived from OGC 07-097; RM-OA 2007]
Session	Temporary association between a subject and a principal as a result of an authentication process initiated by the subject. Information about a session is stored in authentication session information. (OGC 07-097; RM-OA 2007)
Software Architecture	The structure or structures of an information system consisting of entities and their externally visible properties, and the relationships among them [SOA-RM, 2006]
Software Component	Software program unit that performs one or more functions and that communicates and interoperates with other components through common interfaces. [derived from http://www.opengeospatial.org/resources/?page=glossary]
Source System	Container of unstructured, semi-structured, or structured data and/or a provider of functions in terms of services. The source systems are of very heterogeneous nature and contain information in a variety of types and formats.
Spatial Context	Specification of a spatial location of an observed property determined by a combination of a point, a line, an area, a volume and/or a vector field.
Spatial Data Infrastructure	Relevant base collection of technologies, policies, and institutional arrangements that facilitate the availability of and access to spatial data. The Spatial Data Infrastructure provides a basis for spatial data discovery, evaluation, and application for users and providers within all levels of government, the commercial sector, the non-profit sector, academia and by citizens in general.[http://www.gsdi.org/pubs/cookbook/chapter01.html#spatial]
Subject	Abstract representation of a user or a software component in an ORCHESTRA Application.
System	Something of interest as a whole or as comprised of parts. Therefore, a sys-

	tem may be referred to as an entity. A component of a system may itself be a system, in which case it may be called a subsystem. [ISO/IEC 10746-2:1996]
System User	Provider of services that are used for an application domain as well as IT architects, system developers, integrators, and administrators that conceive, develop, deploy and run applications for an application domain. [OGC 07-097; RM-OA 2007]
Task Ontology	[To be defined]
Technology Viewpoint	Viewpoint of an ODP system and its environment that focuses on the choice of technology in that system. [ISO/IEC 10746-2]
Temporal Context	Specification of the temporal reference of an observed property based on the absolute time. It can be a single point in time, a time sequence, a time period, or a combination of these. In a sampling system, for example, several time periods and time points are needed to describe the time behaviour. However, a time point is already an abstraction. It refers to a small time interval.
Thematic Service	Service that provides an application domain-specific functionality built on top and by usage of architecture services and/or other thematic services. [Derived from OGC 07-097; RM-OA 2007]
Thesaurus	Synonym and antonym repository for data vocabulary terminology.
Transaction	Transaction is a feature of the architecture that supports the co-ordination of results or operations on state in a multi-step interaction. The fundamental characteristic of a transaction is the ability to join multiple actions into the same unit of work, such that the actions either succeed or fail as a unit. [W3C, http://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/#transaction]
Universe of discourse	View of the real or hypothetical world that includes everything of interest. [ISO 19101]
Viewpoint	Form of abstraction achieved using a selected set of architectural concepts and structuring rules, in order to focus on particular concerns within a system. [ISO/IEC 10746-2]
Web Service	Self-contained, self-describing, modular service that can be published, located, and invoked across the Web. A Web service performs functions, which can be anything from simple requests to complicated business processes. Once a Web service is deployed, other applications (and other Web services) can discover and invoke the deployed service.
Workflow	Automation of a business process, in whole or part, during which documents, information, or tasks are passed from one participant to another for action, according to a set of procedural rules. [ISO/DIS 19119]
Workflow Engine	[To be defined]

<end of document>