UNIVERSITÀ DI PISA

Scuola di Dottorato in Ingegneria "Leonardo da Vinci"

Corso di Dottorato di Ricerca in
INGEGNERIA DELL'INFORMAZIONE

Tesi di Dottorato di Ricerca

Science 2.0 Repositories

Autore:

Massimiliano Assante ________________

Relatori:

Prof. Cinzia Bernardeschi ________________

Dr. Pasquale Pagano ________________

Anno 2015

SSD ING-INF/05

Sommario

I continui avanzamenti dell'information and communication technology (ICT) nel campo delle infrastrutture di ricerca stanno cambiando continuamente il modo in cui la ricerca e la comunicazione scientifica vengono effettuate. Scienziati, finanziatori, ed organizzazioni stanno cambiando, e muovendo, i modelli di pubblicazione della ricerca ben aldilà degli articoli tradizionali. Lo scopo è quello di seguire un approccio olistico, dove pubblicare include qualsiasi prodotto (e.g. pubblicazioni, dati, esperimenti, software, siti web, blogs) risultante da un'attività di ricerca e rilevante per l'interpretazione, la valutazione, ed il riuso di essa o di una sua parte.

L'implementazione di questa visione, oggi, è sostanzialmente ispirata ai workflow di comunicazione scientifica di letteratura, che separano il "posto" dove la ricerca viene effettuata dal "posto" dove questa viene pubblicata e condivisa. In questa tesi si sostiene che questo modello non possa adattarsi bene alle pratiche di comunicazione scientifica previste nel contesto della Science 2.0; pertanto si propone una nuova classe di data repository scientifici, denominata "Science 2.0 Repositories" (SciRepo), come possibile soluzione. Gli SciRepo affrontano gli ostacoli legati alla pubblicazione di dati rimuovendo la distinzione fra ciclo di vita della ricerca, e pubblicazione della ricerca. Essi consentono workflow di comunicazione scientifica effettivi, permettendo che la creazione e la pubblicazione di prodotti di ricerca avvenga "all'interno" dell'infrastruttura di ricerca, e "durante" le attività di ricerca.

Sono presentati altresì i benefici risultanti dal pubblicare attraverso SciRepo e se ne propone un modello di riferimento. Si definisce inoltre un'architettura di riferimento per una SciRepo platform, una piattaforma general-purpose che facilita la realizzazione di uno SciRepo su di un qualsiasi ambiente ICT di un'infrastruttura di ricerca, con effort e costi limitati, se confrontati con gli approcci da zero.

Abstract

Information and communication technology (ICT) advances in research infrastructures are continuously changing the way research and scientific communication are performed. Scientists, funders, and organisations are moving the paradigm of Research Publishing well beyond traditional articles. The aim is to pursue an holistic approach where publishing includes any product (e.g. publications, datasets, experiments, software, web sites, blogs) resulting from a research activity and relevant to the interpretation, evaluation, and reuse of the activity or part of it.

We show that the implementation of this vision is today mainly inspired by literature scientific communication workflows, which separate the “where” research is conducted from the “where” research is published and shared. We claim that this model cannot fit well with scientific communication practice envisaged in Science 2.0 settings and propose a new class of scientific data repositories, denominated “Science 2.0 Repositories” (SciRepos), as a possible solution. SciRepos address the data publishing issues by blurring the distinction between research life-cycle and research publishing. They enable effective scientific communication workflows by making research products creation and publishing occur “within” the research infrastructure and “during” the research activities.

We present the benefits resulting from Publishing in SciRepos and define a Reference Model and a Reference Architecture for a SciRepo platform, a general purpose platform facilitating the realisation of a SciRepo over any ICT-based research infrastructure environment with limited costs and efforts if compared with from-scratch approaches.

to Elisa

Contents

List of Figures

List of Tables

1

Introduction

> Science, my lad, is made up of mistakes, but they are mistakes which it is useful to make, because they lead little by little to the truth.
>
> Jules Verne

1.1 Motivation

We began this research work with the objective of taking the latest "ICT world" advances, exploited mainly for industrial and commercial purposes into the scientific publishing domain. The ICT world *has modernised* how data get stored, accessed and shared, *has gone well beyond* expectations in terms of data processing scalability, elasticity, velocity, and data resiliency, *has changed* the way information is exchanged and delivered. A small revolution that is having difficulties in finding its way into the scientific communication world. In fact, although during the last decade, these ICT advances have changed the way research is conducted within Research Infrastructures[1] (RIs), Research Publishing is still bounded to its traditional practices. Research Infrastructures enabled a remarkable increase of scientific production, including *(i) data intensive science*, i.e., the availability of datasets at petabyte level, processed through simulation software and empowered by high performance computing, *(ii) open science*, i.e., access to scientific data as well as reliability of scientific

[1] A Research Infrastructure is intended as the compound of elements regarding the organisation (roles, procedures, etc.), the structure (buildings, laboratories, etc.), and the technology (microscopes, telescopes, sensors, computers, internet, applications, etc.) underpinning the implementation of scientific research.

discovery, and *(iii) collaboration science*, i.e., a changing paradigm towards open research collaboration enabling large-scale, remote collaboration of scientists with the use of internet-based tools.

Thus, if on one hand Research today is based on digital research products, such as datasets, software, and services, and the access and sharing of such products has mutated in order to adapt the underlying business models and mission to such new scenarios, on the other hand Research Publishing is still adopting the traditional article paradigm, which clearly, cannot cope with the increasing demands of immediate access and effective reuse of research results.

Scientists, funders, and organisations are therefore pushing for innovative scientific communication workflows (deposition, quality assessment and dissemination), marrying an holistic approach where publishing includes in principle any product (e.g. publications, datasets, experiments, software, web sites, blogs) resulting from a research activity, that is relevant to the interpretation, evaluation, and reuse of the activity or part of it.

Nowadays, the implementation of this vision is mainly inspired by scientific literature communication workflows, which separate *the place where research is conducted*, i.e., RIs, from *the place where research is published and shared*, i.e., third-party marketplace services. Specifically, research products are published “elsewhere” and “on date”, i.e., when the scientists feel the products obtained so far are sufficiently mature.

In our opinion this model cannot fit well when other kinds of research products are involved, for which effective interpretation, evaluation, and reuse are needed. We argue that to enable effective scientific communication workflows, research product creation and publishing should both occur “within” the RI (as opposed to “elsewhere”) and “during” the research activities (as opposed to “on date”). To facilitate this, research infrastructure ICT services should not only be conceived to provide scientists with facilities for carrying out their research activities. Rather, they should also support marketplace like facilities, enabling RI scientists to publish products created by research activities and other scientists to discover and reuse them. Strictly speaking, RIs should not rely on third-party marketplace sources to publish their products, they should integrate them into the RI.

Unfortunately, current repository services, which are key elements of the research marketplace, are not suitable to implement this vision, as they are designed not to integrate with existing RI ICT services but instead to support today’s notion of the “elsewhere” and “on date” research marketplace.

In this dissertation we present the state-of-the-art of the publishing practices in Science 2.0[2], which comprises approaches for enhancing traditional publications and research data publishing. We critically review these approaches and survey those promoted by scientific data repositories for data publishing, as these repositories have a key role in the data publishing picture, i.e., to implement data stewardship practices and to foster scientific datasets collection, curation, preservation, and dissemination. We then identify a set of drawbacks affecting the current publishing practices and a number of obstacles hindering modern scientific communication workflows, demonstrating that their implementation separate the “where” research is conducted from the “where” research is published and shared.

We propose an innovative class of repositories: Science 2.0 Repositories (SciRepos), that, living in synergy with RIs, meet research publishing requirements arising in Science 2.0 by blurring the distinction between research life-cycle and research publishing. SciRepos interface with the ICT services of research infrastructures to intercept and publish research products while providing researchers with social networking tools for discovery, notification, sharing, discussion, and assessment of research products.

We continue reporting on the scientific communication workflows as realised by SciRepo and discuss the benefits of research publishing using it. We define a Reference Model and a Reference Architecture for a SciRepo platform, a general purpose platform facilitating the realisation of a SciRepo over any ICT-based research infrastructure environment with limited costs and efforts if compared with from-scratch approaches.

1.2 Research Contributions

The main innovative research contribution presented in this work is the introduction in literature of the *notion of Science 2.0 Repositories* (SciRepos) (cf. Sec. 3.1), repositories tailored to serve the Science 2.0 vision. Plenty of articles in literature are discussing and proposing ways to move beyond the traditional article publication paradigm but none of them, according to our knowledge, has the ambition to cope with the entire scientific literature communication workflow and to drastically

[2] Science 2.0 describes the on-going evolution in the modus operandi of doing research and organising science. These changes in the dynamics of science and research are enabled by digital technologies and driven by the globalisation of the scientific community, as well as the need to address the Grand Challenges of our times. They have an impact on the entire research cycle, from the inception of research to its publication, as well as on the way in which this cycle is organised [86].

enhance it while keeping it affordable and preserving the way scientists perform research. Even if the SciRepo approach, tackling research and publication needs as a unicum did not exist yet in literature, part of the research work has not been done starting from scratch. Rather, it valued the experience and the knowledge acquired by the ISTI-CNR through the realisation and experimentation of real data management systems [33, 10, 9].

In order to get a full understanding of the problem space we performed a *Survey on Scientific Data Repositories*, this survey (cf. Sec. 2.1.2) is indeed a research contribution of this dissertation, and it has been carried out by analysing the repository websites, by looking for literature about them, by (programmatically) harvesting the repositories for their metadata and by contacting repository maintainers when needed.

Finally, SciRepos is not only presented as an innovative approach but its theoretical foundations are presented throughout two main research contributions, a *(i) Reference Model for a SciRepo Platform.* i.e. an abstract framework for understanding the significant concepts and relationships among the components of SciRepos. We therefore defined the core of a model that represents the significant entities and relationships of a SciRepo (cf. Sec. 3.2), and a *(ii) Reference Architecture for a SciRepo Platform.* an abstract framework for describing the components required by a SciRepo, their relations and patterns. Useful for developing consistent services that support them (cf. Chap. 4).

The presentation of the research contributions outlined above is organised as described in the next section.

1.3 Outline of Dissertation

This dissertation is organised in five chapters. Chapter 1 introduces this dissertation by outlining the problem space and the motivations.

Chapter 2 presents research publishing in Science 2.0 as is today. We analysed three main approaches within this domain: the approach for moving beyond traditional articles, namely Enhanced Publications, and the two approaches for data publishing promoted by Data Journals and by Scientific Data Repositories. Specifically, for the latter, we performed a Survey on their state of the art offering, presenting how the current solutions try to overcome data publishing problems. A key findings section identifies a set of barriers to modern scientific communication and the weaknesses in the current way of publishing.

Chapter 3 claims that the current publishing model cannot fit well with scientific communication practices envisaged in Science 2.0 settings and present Science 2.0

Repository (SciRepo) as a possible solution. The chapter continues presenting a Reference Model and concludes reporting on the benefits of research publishing using SciRepo.

Chapter 4 presents a SciRepo Platform Reference Architecture, a template model that maps the functionalities defined in the Reference Model onto software components that implement them. It indeed comprises directions for architecture principles and best practices providing an architecture baseline and an architecture blueprint.

Chapter 5 concludes this dissertation by reporting a summary on the realised achievements.

2

Research Publishing in Science 2.0

In this chapter we present the state of the art of publishing practices in Science 2.0, which embraces approaches for enhancing traditional publications and research data publishing. The chapter also contains a Survey performed on the the state of the art offering of Scientific Data Repositories.

A key findings section concludes by critically review these approaches and by reporting on the identified drawbacks affecting the current publishing practices.

2.1 State of the Art

Research Publishing is the activity of making research work available to other scientists or experts in a given publishing field. Research work is generally published in scientific books, journal articles, conference proceedings and thesis, while the part of research work that is not formally published (but solely printed or put up over the Internet) is typically called “grey literature”.

When referring to the action of “publishing”, most people would refer to the scientific communication practices that are typical of research literature. These practices are *(a)* supported by policies and services of a “research marketplace”, intended as the set of online services thanks to which publications can be shared (e.g., discovered, accessed, cited, referred, interlinked, tagged) by scientists, *(b)* applied to *selected* research products while research activity is still ongoing, i.e., it is up to the scientists involved in a research activity to decide “what” is a candidate research product and “when” to publish it.

Publishing consists of the following phases:

1. **Deposition:** scientists deposit research products into accredited repositories (e.g., Institutional repositories, Journal’s repositories);
2. **Quality assessment:** scientists submit their candidate products to a peer-review process of some kind (e.g., single/double blind, open peer review);
3. **Dissemination:** besides repository-provided dissemination tasks, there are a number of web applications (e.g., Google Scholar, DBLP) taking care of aggregating, indexing, and cataloguing publication metadata, in order to provide advanced publication discovery mechanisms and citation indexes.

As shown in Figure 2.1, publishing is usually conceived as the concluding step of the research activity life-cycle. It comes conceptually after the research activity step - i.e. the phase leading to the production of research results - although this does not imply that this step is complete. It is expected that a new research activity lifecycle starts by using the results of previous life-cycles manifested in published products.

Research work is mainly carried out through Research Infrastructures (RI), intended as the compound of elements regarding the organisation (roles, procedures, etc.), the structure (buildings, laboratories, etc.), and the technology (microscopes, telescopes, sensors, computers, Internet, applications, etc.) underpinning the implementation of scientific research. Indeed RIs are the setting supporting scientists at performing their research activities, which generally consist in running experiments relying on existing research products (e.g., publications, datasets, software, manuals, services, processes, web sites, blogs) in order to yield new research products.

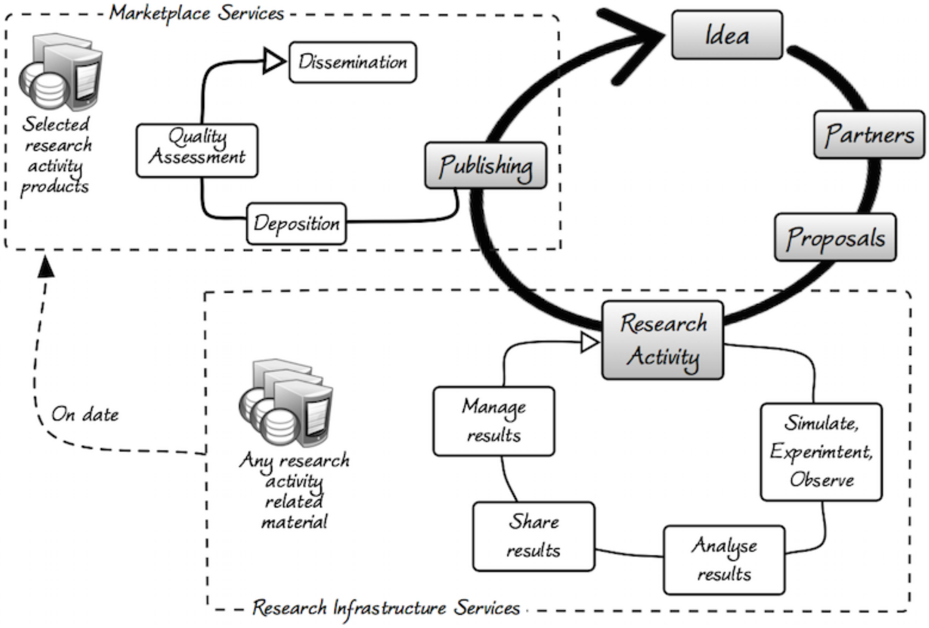


Figure 2.1: Publishing in a Research Lifecycle

In such scenarios, ICT services are becoming increasingly essential to perform research activities. They may range from simple computers and connection to the internet (e.g., web and email) to data centres offering computational resources (e.g., web servers), services for data management (e.g., document stores, column stores) and processing (e.g., workflow management).

ICT services are intended not only for supporting scientific investigation, but also for publishing and re-using the resulting research products. As a matter of fact, these days scientific communication workflows are based on the availability of Internet connection and devices, which make drafting, publishing, and accessing scientific publications in digital form the norm for the average scientists. Besides, ICT services have been playing a central role in shaping up modern forms of scientific communication, which are today reaching beyond publishing articles in digital format. For example, many RIs provide scientists with ICT tools for the elaboration of large quantities of data, and the community invest energies into collecting, curating, and creating research data.

The recent advances in the Research Publishing domain can be clustered in two classes. The first class is represented by *Research data publishing*, a practice where research data are made public as to enable their reuse as well as attribution and credit for the data producer. Data publishing elevates research datasets to primary research products on par with papers. There are at least two approaches in literature characterising this publishing domain, publishing *via Data Journals* and publishing *via Scientific Data Repositories*.

The second class representing recent advances in the Research Publishing domain is embodied by the *Enhanced Publications*, a practice aiming at overcoming the traditional research publishing paradigm limits, based on the general and growing awareness of the importance of publishing research results together with their experimental context.

In the following we present these two classes at their current state-of-the-art, surveying and analysing different literature definitions, suggested by researchers and organisations spread across the globe.

2.1.1 Research Data Publishing via Data Journals

The issues related to formally publishing and citing datasets are unquestionably enumerated and discussed by Lawrence et al. [62] in 2011 and by Callaghan et al. [31] in 2012. Specifically, the latter article reports on how the authors developed and formalised a method for formally citing and publishing the datasets stored in NERC[1]'s environmental data centres. They discuss the differences between informal research data publishing and formal research data publishing. For the former they state:

> "*It is now possible to publish data relatively easily; at its most basic all a researcher has to do is to stick the files on a website somewhere. This makes the data open, but without any form of long-term commitment. There are no guarantees that the data will still be there in six months, or that the files won't get corrupted.*"

and continue by moving from a preservation issue to a documentation one for helping proper interpretation and reuse:

> *"Furthermore, it is possible that a scientist who isn't the data creator won't be able understand the contents or even open the files at all. Even if the dataset is readable and has sufficient metadata, there is no information about the scientific quality of the dataset, other than that attached to the creator's reputation.*"

while for the latter, formal publishing:

> "...*a formal Publishing process adds value to the dataset for the future consumers of the data. This may be by providing an indication of the scientific*

[1] NERC: Natural Environment Research Council is the UK's biggest funder of environmental research and training, http://www.nerc.ac.uk

> *quality and importance of the dataset (as measured through a process of peer-review), or by ensuring that the dataset is complete, frozen, and has enough supporting metadata and other information to allow it to be used by others in the years to come. Publishing implies a commitment to persistence of the data. It also provides a mechanism for allowing data producers to obtain academic credit for their work in creating the datasets.*"

They conclude the study with recommendations on data citation and publication arguing that the aim is to have datasets as a "*first class research output*" that will be available, peer-reviewed, citable, easily discoverable and reusable. Their proposed plans for data publication identified the *Data Paper* solution:

> *"data publication involve working with academic publishers to develop a new style of article: a data paper, which would describe the dataset, providing information on the what, where, why, how and who of the data. The data paper would contain a link back (a DOI) to the dataset in its repository, and the journal publishers would not actually host the data"*.

As a follow up, a number of initiatives have started to realise *Data Journals*, i.e., journals containing data papers, in various domains ranging from archeology to chemistry, ecology and oceanography. Candela et al. surveyed them in 2014 [32]. From their study, 15 different publishers publishing 116 data journals have been identified. They grouped these data journals into sets corresponding to publishers, the result is given in Table 2.1.

The survey analyses data journals from different dimensions. Specifically, the nature of data journals and the number of *(i)* published journals by topic, *(ii)* published papers and operational data journals, by year, *(iii)* data journals indexed by Thomson Reuters, and *(iv)* open access data journals. They conclude the study by reporting that:

> *"...data journals are now an established phenomenon in the scientific literature. In fact, the number of published data papers and data journals is rapidly growing. The 23.5% of the existing data papers has been published in 2013. The majority of data journals (69.82%) is indexed by well known professional services, namely Thomson Reuters Web of Science"*.

It is important to note that Data Journals play also an important role in dataset documentation. Data papers are envisaged to play well the role of "core documentation" for humans. It is expected that they contain a link to the dataset as well as that the dataset contain a link back. It is possible to envisage that many data papers are actually produced to properly communicate the availability of the same dataset. Each of

Publisher	Number of Journals associated
BioMed Central	85
Chemistry Central	3
Pensoft Publishers	7
SpringerOpen	8
Ubiquity Press	3

Table 2.1: Data Journals number grouped by Publisher.

such data papers is oriented to a target audience. It is up to the dataset publisher to figure out what are the primary communities potentially interested in the dataset and provide them with a specific data paper in community journals. This does not infringe the willingness to publish a dataset for any community, rather it helps in producing potentially better documentation.

Data Journals promote also approaches for data preservation. In 2013 Burda and Teuteberg [27] offered a review of the literature on digital preservation analysed under a specific point of view, i.e., they conceive data preservation as an organisational issue of decision-making, concerning what (and what not) to preserve while considering both organisational constraints such as costs or compliance objectives and technological aspects. Their investigation is mainly focused on literature from the area of computer science and management information systems, finding that that digital preservation has gained little attention in management information systems research compared with the computer science discipline. Based on this finding, they propose an agenda for future research aiming at the construction of a reference model for digital preservation.

It is worth to discuss the approach promoted by data journals for dataset citation, the practice of providing a reference to data or a dataset intended as a description of data properties that enable credits and attribution, discover, interlinking, and access to it. It results in the availability of data papers, which reconciles the dataset citation practices with the literature citation practices. Further, data papers deals with the citation principles, e.g., they definitely give importance to datasets, provide for credit and attribution enabling authors to clearly specify who is contributed what in the endeavour leading to the dataset, are called to support dataset access. Whenever it is possible to use a literature-like reference for citing a dataset, this can be produced by referring to the data paper associated with the dataset. However, to make this approach working, data journal publishers must agree and develop both detailed guidelines and copy-editing practices to guarantee that the data papers they publish contain an intelligible citation to the dataset(s) each paper is about.

Finally, Data papers are expected to be published concurrently with the publication of the dataset in a repository, thus it is fundamental to strengthen the collaboration and coordination activities between data paper publishers and Scientific Data Repositories, for which we present an extensive survey in the following.

2.1.2 Research Data Publishing via Scientific Data Repositories

A different approach, still belonging to the data publishing research domain, is represented by the diffusion of *Scientific Data Repositories* (e.g., Dryad, FigShare, GigaDB, ICPSR, Pangaea, Zenodo etc.). Scientific data repositories [68] have a key role in science as well as in the data publishing picture. Their role is to foster scientific datasets collection, curation, preservation, long term availability and dissemination. Such repositories are principally, but not only, promoted by disciplines producing vast amount of data, such as physics, genetics, or environmental sciences, for example, SDSS at FermiLab, GenBank at NCBI Data, or the British Atmospheric Data Centre.

Indeed publishers and private firms have started to support scientists to deposit their data into scientific data repositories. This fact is manly supported by two "standard features" most of these repositories offer, namely high-quality preservation and persistent identifiers for scientists data. Scientific Data Repositories can be clustered in two classes: *Open Repositories*, i.e., repositories that anyone can use, despite of institutional affiliation, to store any type of scholarly output, and *Disciplinary Repositories*, i.e., repositories meant to store specialised research data with relevant communities. They offer many of the same features of the previous class, complementing them with special features regarding disciplinary data.

The proliferation of Scientific Data Repositories and the pressing needs for proper research data publishing practices make the time ripe for a systematic review of their state of the art offering, which is given in the following. Our interest is to analyse the characteristics of the scientific data repositories dealing with the data publishing challenge. We analyse the current practices for research data publishing promoted by these repositories focusing on the ones that are generalist, i.e., open to publish any data. This guarantees that the findings are not affected by community or data-type specific aspects and genuinely highlight the challenging issues emerging when dealing with data publishing.

This survey has been performed by analysing the repository websites, by searching the web for literature about them, by (programmatically) harvesting the repositories for their metadata and by contacting repository responsibles when needed.

Repositories Selection

Marcialand Hemminger [68] surveyed scientific data repositories on the web with the aim of identifying a variety of characteristics related to their general nature, their business practices and policies. The survey ends with a lot of recommendations. The authors observe that "Technology has made it easier to develop or start scientific data repositories, but a lot of effort is still required to maintain them"; more, that there are a number of methodological, technical and legal obstacles for reaching the objectives of publishing research data in order to make data discoverable and reusable, and data owners and custodians get the recognition they deserve for making datasets public.

	Type	Founded	Country	Software
3TU.Datacentrum	Institution	2008	Netherlands	In-house
CSIRO DAP	Institution	2011	Australia	In-house
Dyad	Organization	2008	United States	DSpace
Figshare	Company	2011	United Kingdom	In-house
Zenodo	Organization	2013	Switzerland	Invenio

Table 2.2: Scientific Data Repositories studied.

A plenty of scientific data repositories currently exists. Directories of research data repositories enumerate a very large and continuously increasing number of repositories, most of them being disciplinary ones, with very specific characteristics. We focus on repositories qualifying themselves as "generalist" , i.e., open to publish any data. This choice allows avoiding any "bias" or "noise" resulting from the peculiarities of specific disciplines and communities. As a matter of fact, although an important strength of research data repositories may be its domain specificity, the multidisciplinary nature of modern science calls for unifying environments facilitating open and easy data sharing [68]. Further, the focus on "generalist" repositories allows to distill the real state of the art and the open issues faced when dealing with the possibly open ended set of data typologies and communities characterising the "long-tail" of science i.e., science usually conducted by individual investigators however globally producing a large volume of distributed high-value information.

As data journals are emerging to address the data publishing challenge by relying on recommended repositories for the data publication (cf. Sec. 2.1.1), we have included in our sample of repositories to be surveyed those recommended by existing data journals. This choice somehow makes it possible to affirm that the selected repositories have an acknowledged role in the data publishing practice recognised by the research community.

This process has led to the identification of the following five repositories:

- **3TU.Datacentrum** originated from a cooperation of the three technical universities in the Netherlands (Delft University of Technology, Eindhoven University of Technology, and University of Twente). Its aim is to provide the scientific community with a sustainable and persistent archive for research data. The management is supported by the TU Delft Library.

  Website: `http://datacentrum.3tu.nl`

- **CSIRO Data Access Portal** was established by the Australia's national science agency CSIRO (Commonwealth Scientific and Industrial Research Organisation). Its aim is to manage, discover and share data across different research fields; the repository is supported by CSIRO, and is a partner in the Australian National Data Service (ANDS).

  Website: `https://data.csiro.au`

- **Dryad** was born as an initiative of a group of journals aiming to adopt a joint data archiving policy (JDAP) for their publications, as well as to set up a community-governed infrastructure for data archiving and management. The repository is supported by a nonprofit membership organisation, including journals and publishers, scientific societies, research institutions and libraries, and research funding organisations.

  Website: `http://datadryad.org`

- **Figshare** was started by Mark Hahnel, an Imperial College PhD student passionate about open data, as a way to store, manage and freely disseminate any kind of research outputs. The repository is one of the products of Digital Science, a technology division of the “Macmillan Science and Education” company, working to make research more efficient.

  Website: `http://figshare.com`

- **Zenodo** was launched within the EU FP7 project OpenAIRE-plus [66], as part of a European-wide research infrastructure, with the aim to enable researchers to preserve and share any kind of research output, with a particular predilection for the long-tail of science. The repository is hosted by the European Organisation for Nuclear Research (CERN) and funded by the European Commission via OpenAIRE (openaire.eu/).

  Website: `http://zenodo.org`

Table 2.2 gives some basic data on the selected repositories including the type, the year of foundation, the country, and the underlying software.

Some indicators on the "distributions" of the datasets published by these repositories are given in Table 2.3 and Table 2.4. Table 2.3 reports the number of dataset items published by repositories per year. From this tables it emerges that Figshare is the larger repository in the sample, it holds more than the 96% of the items published. From Table 2.3 we extracted the chart reported in Figure 2.2, showing the increment in the number of dataset items published by repositories through the years until 2014, our metadata-harvesting programs in fact harvested the repositories setting as superior limit (not included) the first of January 2015. From this chart it clearly comes out that the publishing dataset connected to publications is gaining momentum.

Table 2.4 reports the most frequent subjects associated with published datasets by repositories. In this table we also report the number of different subjects found per repository in the last row, FigShare adopts a controlled dictionary for its subjects and this is why only 164 are reported.

	up to 2009	2010	2011	2012	2013	2014
3TU.Datacentrum	1,356	337	451	371	151	40
CSIRO DAP	0	0	60	64	462	481
Dyad	164	230	804	1,253	2,050	2,536
Figshare	0	0	16,929	28,224	108,234	94,365
Zenodo	1	1	1	2	39	236
Total	1,521	568	18,245	29,914	110,936	97,658

Table 2.3: Datasets published by Scientific Data Repositories.

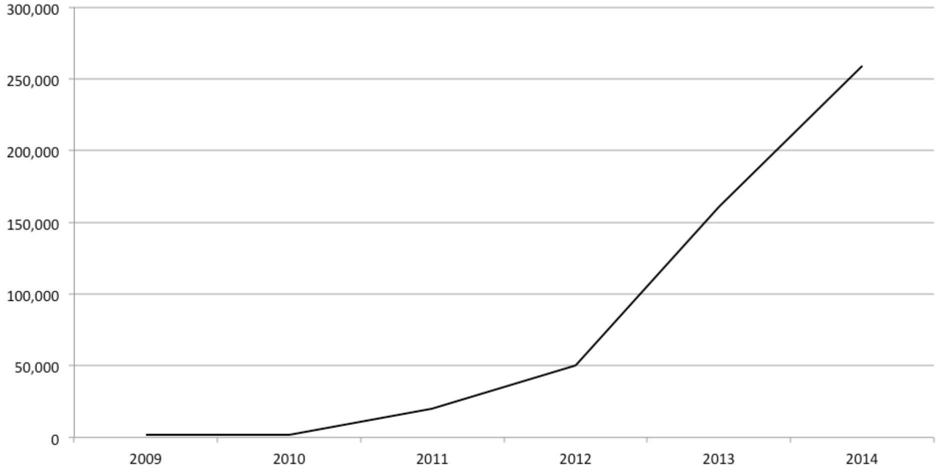


Figure 2.2: Increment of the number of dataset items published by repositories through years

Repositories Analysis

Although the term data publishing is still subject to debates [60][81], there is an agreement on some essential aspects that characterise it. Publishing data calls for datasets to be: *(i)* properly documented, *(ii)* formatted, *(iii)* available, *(iv)* discoverable, and *(v)* formally citable. In addition to that, there are two aspects complementing the data publication picture: *(vi)* when and to what extent published data must be validated, and *(vii)* data publishing costs, for all the actors involved in the process. In this study we use the term “dataset” to refer to the information unit subject of the data publishing activity, no matter how many files it materialises. This term includes the term “data package” adopted by Dryad to define a set of data files associated with a publication; it also includes both “dataset” and “fileset” adopted by Figshare to indicate raw data (the former) and a group of multiple files citable as a single object (the latter).

In the next we analyse the repository practices focusing on the above seven aspects.

Dataset Documentation

A published dataset is of no value if it is not accompanied with appropriate documentation describing aspects like what the dataset is and how it has been obtained. Very often this documentation is known as metadata, part of it is bibliographic metadata while the part oriented to actually promote reuse is known as data provenance [98, 35]. The “quality” of such a documentation is equally important as the “quality”

	3TU. Datacentrum	CSIRO DAP	Dryad	Figshare	Zenodo
#1	N/A (2400 – 71.4%)	20199 (503 – 6.3%)	adaptation (432 – 1.3%)	Biological Sciences (119,330 – 22%)	N/A (56 – 6.4%)
#2	DTS (38 – 1.4%)	pulsars (237 – 3%)	N/A (427 – 1.2%)	Medicine (58,137 – 10.7%)	web crawling (34 – 3.9%)
#3	s. water temp. (37 - 1.1%)	neutron stars (233 – 2.9%)	pop. gen. (375 – 1.1%)	Genetics (32,012 – 5.9%)	web domains (34 – 3.9%)
#4	530 Physics (23 - 0.7%)	Australia (215 – 2.7%)	Speciation (278 – 0.8%)	Biotechnology (27,331 – 5%)	ind. ds. (27 – 3.1%)
#5	apsp (19 - 0.6%)	pulsar (120 – 1.5%)	Ecological Genetics (235 – 0.7%)	Infectious Diseases (20,781 – 3.8%)	backlinks (24 – 2.8%)
#6	comp. sci. (19 - 0.6%)	climate change (113 – 1.4%)	Phylo-geography (211 – 0.6%)	Biochemistry (19,450 – 3.6%)	neuroanatomy (10 – 1.2%)
#7	stp (19 - 0.6%)	alien plant (108 – 1.4%)	Hybridization (192 – 0.6%)	Ecology (18,003 – 3.3%)	indexing system (10 – 1.2%)
#8	stn (19 - 0.6%)	naturalised plant (108 – 1.4%)	Conservation Genetics (187 – 0.5%)	Cell Biology (17,933 – 3.3%)	drosophila (9 – 1%)
#9	shortest path (19 - 0.6%)	spd (108 – 1.4%)	Insects (179 – 0.5%)	Neuroscience (17,182 – 3,2%)	linguistics (8 – 0.9%)
#10	ozone installations (18 - 0.5%)	world (108 – 1.4%)	Fish (147 – 0.4%)	Chemistry (14,976 – 2.8%)	open data (8 – 0.9%)
Distinct Subjects	457	1,416	14,426	164	513

The following subjects are shortened: 'stream water temp.' is 's. water temperature'; 'pop. gen. - empirical' is 'population genetics - empirical'; 'comp. sci.' is '000 computer science, knowledge & systems'; 'apsp' is 'apsp, all-pairs shortest paths'; 'stn' is 'stn, simple temporal network'; 'stp' is 'stp, simple temporal problem'; 'spd' is 'species distribution model'; 'ind. ds' is 'indipendent data source'.

Table 2.4: Top 10 subjects associated with published datasets

of the dataset per se [94, 26], since this documentation is the enabler for dataset "re-use". To identify the metadata supported by the selected repositories, we have analysed the metadata collected at submission time and the metadata exposed at visualisation time, i.e., when a repository user is accessing the dataset landing page. The following 10 classes of attributes have been identified:

***Availability**:* attributes promoting access to dataset, namely a DOI;

***Bibliographic description**:* attributes typically associated with a scholarly publication such as title, authors, keywords, brief description or abstract;

***Bibliometric data**:* attributes reporting dataset publication statistics including number of data visualisations, downloads of the dataset, including research goals, type of and shares;

***Coverage**:* attributes describing the dataset "extension", including spatial, temporal and taxonomic coverage;

***Date**:* attributes providing information about "when" the dataset was created, submitted and published, including embargo period;

***Format**:* attributes characterising the dataset from the formatting perspective including the size;

***License**:* attributes describing the policies ruling the dataset reuse, including access rights and licenses;

***Paper reference**:* attributes providing a reference to related publications, including a DOI or a URL;

***Project**:* attributes describing the initiative leading to the production of the dataset, including research goals, type of research and funding sources;

***Provenance**:* attributes specifying the methodologies leading to the production of the dataset, including original sources, instruments and software tools used to create the data files.

A summarised picture of the classes of metadata attributes supported by each repository is given in Table 2.5. Repositories make different choices implementing these classes. There is a large heterogeneity with respect to how many attributes repositories support per class, whether the attributes are mandatory or not, whether the attributes are filled by using controlled vocabularies or free text. For example, paper reference information is envisaged by all the selected repositories, but it is mandatory for Dryad only since this repository only accepts datasets associated with

	3TU.Dat.	CSIRO	Dryad	Figshare	Zenodo
Availability	x				x
Bibliographic	x	x	x	x	x
Bibliometric			x	x	
Coverage	x	x	x		
Date	x		x	x	x
Format	x		x	x	
License		x			x
Paper reference	x	x	x	x	x
Project	x	x			x
Provenance		x	x		

Table 2.5: Dataset attributes supported by Scientific Data Repositories.

a scientific publication. Further, Dryad requires very specific attributes about publications associated with the dataset, including title, authors, journal, abstract, keywords and coverage; instead, Zenodo only suggests to provide information about journal name and issue, and the relationship between the dataset and the publication.

At the moment of submission, repositories may ask submitters to upload supporting documentation also. This is the case of CSIRO DAP and Dryad, that encourage authors to provide additional documentation in the form of “ReadMe” files for helping proper interpretation and reuse of the dataset. Specifically, Dryad recommends the ReadMe be a plain text file containing the following information: *(a)* for each file, a short description of what data are included; *(b)* for tabular data: definitions of column headings and row labels; data codes and measurement units; *(c)* any data processing steps that may affect interpretation of results; *(d)* a description of the associated datasets that are stored elsewhere, if any; and *(e)* contact information for questions.

Dataset Formatting

Dataset formatting is intended as the arrangement of the data according to a certain data format. The notion of format applies to different layers of the dataset representation ranging from the file format, i.e., the way data is encoded in a file, to the content format, i.e., the way data is actually organised. For instance, when dealing with tabular data the file format might be a comma separated value (csv) while the content format is about the columns and their values. Having data appropriately formatted is a pre-requisite for any use of it. In fact, dataset formatting impacts on the capacity of current and future software to import the data and make use of it.

There is no definition of research data that can help in clearly identifying what are the kinds of data that scientific data repositories should manage. Currently, the term research data is intended as a very broad and heterogeneous range of material that is produced during a research activity. As a consequence, selected repositories tend to clarify what are the kinds of data they manage. Besides, all the selected repositories agree on the fact that they are expected to be provided with the datasets by means of files, no data streams nor protocols for on-demand accessing the dataset are considered.

For the types of datasets, repositories have no constraints and declare to accept any dataset underlying a research activity. However, there are some peculiarities. Zenodo and Figshare actually accept any research output including papers and presentations, as a consequence they have a scope exceeding research data only. CSIRO and 3TU.Datacentrum repositories in addition to manage every dataset, specifically manage certain datasets. Currently, the specific datasets supported by CSIRO DAP are three: ATNF Astronomy Observations – pulsar observations from the Australia Telescope National Facility in Parkes; AAHL Microscopy Observations – microscopy images from Australian Animal Health Laboratory; and Sensor Data – empirical data about the natural world coming from CSIRO Sensor Networks. 3TU.Datacentrum manage the datasets belonging to the UNESCO-IHE Institute for Water Education, based in Delft, Netherlands.

For file formats, in general there is no particular restriction: all repositories in our sample may accept datasets in any format. However, authors are often encouraged to submit data in "friendly" formats, namely standard formats that are supposed to be suitable for preservation and reuse. For example, 3TU.Datacentrum provides users with a table including the "preferred formats“ for any type of file, namely the optimal file formats used for long-term preservation of data (cf. Sec. 2.1.2). In general, such formats are non-proprietary and well documented. The CSIRO DAP declares the ad-hoc formats supported for specific datasets. For example, all files composing the ATNF Astronomy Observations datasets are compliant with PSRFITS, i.e., a standard for Pulsar Data Storage.

Detailed information on the most common file formats used by the published datasets is given in Table 2.6. The CSIRO most common files are largely used in astronomy, e.g., ".rf“ is for raw data files (folded data), ".cf“ is for calibration data files, ".sf“ is for raw data files (search mode data), and ".FTp“ is for pulsar total intensity profile (averaged over time and frequency).

Independently of file format, repositories do have some limitations on allowed file sizes. They tend to have an upper bound limit yet are open to negotiate extensions to this limit with additional costs (cf. Sec. 2.1.2). Dryad allows to upload no more than 10GB of material for a single publication; 3TU.Datacentrum supports the upload of data sets up to 4 GB; Zenodo currently accepts files up to 2GB although it reports that

	3TU. Datacentrum	CSIRO DAP	Dryad	Figshare	Zenodo
#1	application/x-netcdf (2.353 - 76.77%)	application/fits | sf (139,731 - 27.89%)	Text file | txt (3,759 - 17.24%)	Microsoft Excel | xls (212,773 - 85.88%)	unknown | sav (1,562 - 30.41%)
#2	application/zip (514 - 16.77%)	image/png | png (83,441 - 16.66%)	Microsoft Excel 2007 | xlsx (2,376 - 10.9%)	application/ pdf | pdf (19,724 - 7.96%)	unknown | txt (888 - 17.29%)
#3	text/plain (57 - 1.86%)	application/fits | rf (82,550 - 16.48%)	CSV file | csv (2,012 - 9.23%)	application/ msword | doc (3878 - 1.57%)	unknown | png (815 - 15.87%)
#4	application/octet-stream (27 - 0.88%)	application/fits | FTp (81,533 - 16.28%)	application/ zip | zip (1,688 - 7.74%)	application/ msword | docx (3491 - 1.41%)	unknown | fits (802 - 15.61%)
#5	application/x-hdf5 (22 - 0.72%)	application/fits | cf (76,248 - 15.22%)	Microsoft Excel | xls (1,659 - 7.61%)	application/ vnd.ms-excel | xlsx 1,633 - 0.66%	unknown | ods (199 - 3.87%)
#6	application/x-gzip (17 - 0.55%)	null | adf (9,833 - 1.96%)	not found | not found (1,599 - 7.34%)	image/jpeg | jpg (1,385 - 0.56%)	unknown | zip (195 - 3.8%)
#7	text/xml (16 - 0.52%)	null | dat (4,920 - 0.98%)	Text file | nex (970 - 4.45%)	application/ zip | zip (857 - 0.35%)	unknown | csv (97 - 1.89%)
#8	video/x-msvideo (10 - 0.33%)	null | nit (4,911 - 0,98%)	PDF | pdf (901 - 4.13%)	application/ vnd.ms-excel | xls (567 - 0.23%)	unknown | gz (92 - 1.79%)
#9	video/mpeg (9 - 0.29%)	image/tiff | tif (3,660 - 0.73%)	application/x-fasta|fas (556 - 2.55%)	text/plain | csv (517 - 0.21%)	unknown | m (59 - 1.15%)
#10	application/gzip (8 - 0,26%)	null | 001 (1,637 - 0,.33%)	application/x-fasta|fasta (535 - 2.45%)	text/plain | txt (293 - 0.12%)	unknown | xml (56 - 1.09%)
Distinct Formats	33	1,761	662	403	65

Format is mime type | file extension (where available), The mime type 'Composite Document File V2 Document, No summary info' is shortened into 'Excel 2007'.

Table 2.6: Top 10 formats associated with published datasets

the current infrastructure has been tested with 10GB files. Figshare enables users to store up to 1 GB data in their private space with files up to 250 MB each.

Dataset Availability

Availability is a key feature in data publishing since it aims at guaranteeing that the published data are at consumer's disposal. However, as it happens for other research products, this does not imply that the data must be available for free nor that the use of the data is not constrained or subject to some sort of approval.

For present availability, that actually means access, all the repositories in our sample offer users the possibility to download a dataset as a whole, as well as to download single data files one by one, through apposite links displayed on the dataset web page. For example, CSIRO DAP displays every dataset as a file structure, with checkboxes for selecting only the files to download. Aside from the basic download facility, repositories may provide alternative and customised modalities for accessing the datasets. For example, in addition to standard download facilities CSIRO provides access to datasets via protocols like WebDAV and SFTP as well as makes it possible to registered users to mount the data on selected machines and access the data directly on such machines. In some cases, users are provided with a "preview" of the dataset, e.g., CSIRO make it possible to browse the images in a dataset containing them.

For future availability, repositories have in place both mechanisms aiming at guaranteeing that data are securely archived, e.g., stored in multiple copies, and available over time, e.g., format migration. For secure archiving all the analysed repositories use to store multiple copies of the datasets either on their own premises or on third party service providers, e.g., Figshare uses Amazon facilities, Zenodo uses the CERN Data Centre. In addition, both Dryad and Figshare have partnered with the CLOCKSS organisation, a geographically and geopolitically distributed network of 12 redundant archive nodes located at 12 major research libraries around the world. A failover copy of all contents published by Dryad is maintained in the CLOCKSS Archive, so, if Dryad could no longer maintain the repository as an active service, all Dryad-registered DOIs would be updated to resolve to the copy at the CLOCKSS archive, which would continue to provide free access to the content under the same licensing terms.

For format migration, this is a challenging feature to guarantee because of the almost open ended set of data formats repositories are called to manage (cf. Sec. 2.1.2). Some repositories explicitly declares that they do not guarantee usability and understandability of deposited objects over time, e.g., Zenodo and Dryad. However, Dryad announces to perform format migration, i.e., a new version of a file in a migrated format may be created and added to the original dataset whenever the repository judges that it may facilitate preservation. Migrated files may not contain all the information available in the original file format, but the repository tries to minimise information loss through file format migration. In any case, the information content of the original file is never modified. 3TU.Datacentrum highly stresses the importance of taking appropriate measures to guarantee future accessibility to the data. The License Agreement between 3TU.Datacentrum and data submitters explicitly states that the repository *(i)* shall ensure, to the best of its ability and resources, that the deposited dataset will remain legible and accessible; *(ii)* shall, as far as possible, preserve the dataset unchanged in its original software format, taking account of current technology and the costs of implementation; and, *(iii)* has the right to modify the format of the dataset if this is necessary in order to facilitate the digital sustainability, distribution or re-use of the dataset. 3TU.Datacentrum provides users with a table including the optimal file formats used for long-term preservation of data. The number of supported formats is limited, in order to facilitate future conversions to other formats, but the table is regularly updated with new formats. For each format, the level of support for long-term preservation is indicated as follows:

- **Level 1:** all reasonable actions to maintain usability will be taken; actions may include format migration, normalisation or conversion;
- **Level 2:** limited steps to maintain usability will be taken; file formats may be actively transformed from one format to another to mitigate format obsolescence;
- **Level 3:** only access to the object in its submission file format is provided.

In addition to present and future availability mechanisms, open access and licences contribute to regulate the availability of the published datasets. Two out of the five repositories guarantee unlimited free access, namely Dryad and Figshare. All content published by these repositories is open to the public, so no authorization nor login is required for accessing and downloading data. Dryad may allow authors to put their data under embargo for a limited time, but after that, data will automatically become public. For the other three repositories, access to datasets may be ruled by particular restrictions set by the dataset's owner at submission time; in this case, only authorised users can access and download datasets.

In general, if there is no particular restriction, datasets published by CSIRO DAP and Zenodo are open to the public, while 3TU.Datacentrum datasets are open to every user registered to the repository. 3TU.Datacentrum and CSIRO DAP have defined their own proprietary licenses that users have to comply with. Specifically, the general conditions of use established by 3TU.Datacentrum resemble the Creative Commons Attribution-Noncommercial License, stating that any user wishing to reuse a dataset stored in the repository must always acknowledge the dataset sources and in any case the dataset must not be used for commercial purposes. Similarly, the CSIRO Data License grants user a royalty-free, non-exclusive, non-transferable license for using the dataset only for noncommercial purposes. In alternative, CSIRO allows users to choose a Creative Commons license. For Dryad and Figshare, the Creative Commons CC-Zero License seems to be the best choice. Finally, Zenodo suggests the CC-O License as well, but users are allowed to choose among a wide variety of available licenses.

Dataset Discovery

A published dataset is not very useful if it gets lost in the large multitude of datasets stored into a repository. One of the rationale for publishing a dataset is to improve its “visibility”, to enlarge to potential clients number that will make some use of it.

It is expected that scientific data repositories contribute to the visibility goal by providing their users with a rich array of dataset discovery facilities. These facilities should include user driven facilities, e.g., search and browse, and semi-automatic facilities, e.g., notifications and recommendations. These facilities strongly rely on dataset documentation (cf. Sec. 2.1.2), generally metadata, thus having datasets properly documented has a positive impact on discovery also.

All the repositories analysed in this survey offer the following well known approaches for datasets discovery:

- **Keyword-based search:** users should specify their information need by a set of keywords;
- **Advanced search**: users should specify their information need by a set of field-based filtering criteria. The set of supported fields is repository specific;
- **Browse:** users are provided with a list of datasets to scan. This list can be either the entire list of datasets published by the repository or the list of datasets with respect to a given classification, e.g., keyword, type, format, year, creator. Classifications are repository specific.

CSIRO is the only repository offering two specific searches: *(a)* search by location, i.e., users can specify their information needs by using a map to indicate the area of their interest plus keywords for narrowing the search, and *(b)* collection specific search, i.e., users are provided with forms enabling to specify specific information needs on specific datasets including pulsar observations, microscopy images and sensor data.

In addition to these human-oriented facilities repositories support programmatic access to their content. This is achieved by supporting standard protocols, e.g., OAI-PMH [61], as well as web-based proprietary API. These facilities were exploited by the author to harvest our five selected repositories. In the case of the programmatic API they can also support the deposition phase, thus favouring the integration of the repository in publishing workflows.

The discovery facilities offered by the selected repositories are summarised in Table 2.7.

End-user Facilities					
	3TU.Dat.	CSIRO	Dryad	Figshare	Zenodo
Keyword-based	x	x	x	x	x
Advanced search	x	x	x	x	x
Browse	x	x	x	x	x
Other		x			
Web-based API and Protocols					
	3TU.Dat.	CSIRO	Dryad	Figshare	Zenodo
Harvesting	OAI-PMH	OAI-PMH and REST	OAI-PMH	REST	OAI-PMH
Search		In-house		In-house	

Table 2.7: Dataset discovery facilities

Dataset Citation

Data citation is the practice of providing a reference to data or a dataset intended as a description of data properties that enable credits and attribution, discover, interlinking, and access to. It is the subject of an intense research activity [38]. The most known initiative for data citation is DataCite, an international consortium addressing the challenges of making data sets citable in a harmonised, interoperable and persistent way through the use of persistent identifiers [3][72]. In particular, DataCite supports data centres by providing standards for data publication as well as journal publishers by enabling research articles to be linked to the underlying data.

The repositories in our sample offer various options to address data citation:

- **Citation string:** allow users to get an attribution statement that can be simply copied and pasted in their “documents”. Generally, the citation string has a generic format, e.g., DataCite style [90], consisting of authors, year of publication, title, publisher, repository name, and DOI;
- **Export option:** allows users to directly export the citation to the dataset in a variety of generic formats; the most popular ones are RIS (compatible with software such as EndNote, Reference Manager, ProCite, and RefWorks) and BibTex (com-

patible with software such as LaTeX and BibDesk), but others may be available, e.g., DataCite, Dublin Core, NLM, and MARCXML;

- **Embed option:** allows users to embed a link to the data in the HTML source code of their web pages;
- **Share option:** allows users to share a link to the dataset directly via mail, e.g., Zenodo, or on a variety of social networks, such as Twitter, Facebook, Google Plus, Tumblr, and Mendeley.

The options offered by the analysed repositories are summarised in Table 2.8. It is worth noticing that all these approaches rely on the DOI that is assigned to every dataset at publication time.

	3TU.Dat.	CSIRO	Dryad	Figshare	Zenodo
Citation string	x	x	x	x	x
Export option	x		x	x	x
Embed option	x			x	
Share option			x	x	x

Table 2.8: Dataset citation practices supported by Scientific Data Repositories.

Dataset Validation

Dataset validation is an essential aspect of the data publishing endeavour since it is expected to somehow contribute to assess the “quality” of the dataset. However, it is also among the less defined aspects. With dataset validation it is referred to any process aiming at assessing the “cogency” or “soundness” of the published data. Very often this validation shares commonalities and objectives with the peer review of dataset [26] [70]. Unfortunately, there is no shared understanding of what peer review of datasets mean and what data quality is. The use of peer review in the case of data papers have been discussed in [34]. However, the use of the term peer review for datasets is actually causing some false expectations [81].

For the *pre-publication validation*, apparently all the repositories in our sample perform a validation process of both data and metadata with the only exception of Figshare. However, only Dryad gives accurate information about this process in its Terms of Service. In particular, Dryad curation personnel performs a series of checks ranging from technical one (e.g., files can be open, are not corrupted, do not contain viruses) to administrative ones (e.g., metadata is technically correct, information on the associated paper is in place). In addition, Dryad may review the content for reasons including the presence of inappropriate information and copyright statements incompatible with CC0. In any case, Dryad will not check the dataset from the scientific perspective or modify the content except for accessibility reasons.

As for error checks, 3TU.Datacentrum, Dryad and Zenodo store all datasets along with a checksum of their content. For example, data files submitted to Zenodo are stored with a MD5 checksum of their content, and regular checks of files against their checksums are made.

For the *post-publication validation*, top counts or statistics on datasets download or usage might be considered meaningful, even though this assumption is questionable. Figshare offers the possibility to know how many times each data set has been shared or viewed through its search or browse option. In the first case, information on views and shares are given with the dataset description. With the browse option, it is possible to select a given category from a menu and then produce an ordered list of all the datasets classified in that category. The produced list can be sorted by MostShared or MostViewed. 3TU.Datacentrum publishes aggregated statistics on download datasets as a sort of validation of the centrum itself.

Dataset Publication Costs

One of the factors limiting the publication of datasets is the cost that researchers (data owners) should sustain for making this happening. This cost includes the effort needed to prepare the data to make it possible for others to make use of them, e.g., to document the data, as well as the monetary cost for having the data archived in a trustworthy repository. Incentives and mandates aiming at enlarging the amount of published data have limited impact if they are not accompanied by measures overcoming the publishing cost issue.

The majority of the cost researchers have to sustain when preparing their dataset for publication is out of the scope of repositories mission. In fact, selected repositories do not offer any facility for this but upload of dataset and documentation (cf. Sec. 2.1.2) as well as advices on best practices.

Repositories do have costs for their service operation and tend to cover these costs somehow, some of them by publishing charge.

The Dryad payment model largely resembles the open access model adopted by many journals, as datasets are open to the public, but a charge is always required at the time of data submission. In particular, the repository requires the submitter to pay a Data Publishing Charge (DPC), unless *(a)* the submitter is based in a country classified by the World Bank as a low-income or lower-middle-income economy, or *(b)* the associated journal has already contracted with Dryad to cover the DPC. In order to encourage organisation to cover the DPCs on behalf of their researchers, Dryad offers a series of plans providing for volume discount ranging from a voucher plan, e.g., pay for the publishing of a number of data packages, to subscription plans, e.g., pay an annual fee.

At the time of writing this dissertation, the cost for a single data package is $80. For data packages exceeding the 10GB size limit, $15 will be charged for the first GB and $10 for each additional GB or part thereof. Further, for journals that do not use the integrated data submission service offered by the repository, the submitter will have to pay an additional $10 fee at the time of submission to cover added curation costs.

Dryad is the only repository in our sample that always requires submitters to pay a charge independently of the files size. All the other ones offer at least a minimum storage space where users can publish free of charge. For example, the set of pricing plans offered by Figshare includes a completely free plan, which allows users to store private data up to 1 GB at no charge, with a size constraint of 250 MB per file. The other plans require users to pay a monthly amount depending on the dimension of the private storage space and the file size limit, e.g., a fee of 8$/month guarantees a private space of 10 GB with a size constraint of 500 MB per file. However, payment only involves private storage, as the space for publishing public data is always unlimited for every plan.

Zenodo currently imposes a size constraint of 2 GB per file, as its target is to promote the dissemination of long tail [79]of science for free. However, this repository does not want to exclude larger datasets, so Zenodo is planning to put a ceiling to the space that can be offered at no charge, and introduce payment plans for bigger data.

Finally, no payment is required by 3TU.Datacentrum for datasets up to 4 GB. For larger datasets, users have to contact the repository's staff for arranging a customised upload.

CSIRO DAP is the only “closed“ repository; all the other ones enable external users to register and upload their datasets without any restriction.

Open Issues and Prospects

In the following we discuss the open issues and provide some outlook recommendations about these repositories based on the previous analysis, still focussing on the previously described seven aspects.

Dataset Documentation

One of the major issues affecting documentation approaches is the lack of a systematic and shared way to provide datasets with documentation enabling the actual reuse by both humans and machines. Although a data paper (cf. Sec 2.1.1), if well written, can provide a data user with all the details enabling the potential reuse of the data, it will remain an approach oriented to human users only. The same comment applies to ReadMe-based files.

Publishing a dataset without accompanying it with a bare minimum of documentation making the dataset intelligible for an “average consumer” is a useless exercise bringing no value to science. However, the identification and management of such a documentation is not a sole responsibility of repositories.

Any documentation on the dataset exceeding the data paper(s) yet worth to be published should be managed by the repository. This additional documentation includes any feedback resulting from the actual use of the dataset (cf. Sec. 2.1.2).

Besides the human-oriented documentation, published datasets should be accompanied with proper metadata oriented to serve service providers, i.e., services designed to build on such metadata to offer advanced facilities on the datasets. A set of over 50 metadata schemas for scientific data was analysed by [96] to conclude that *(a)* there are commonalities across disciplines and type of data, and *(b)* there is the need for more metadata-related research to actually increase the access to and reuse of research data. The Research Data Alliance Data Description Registry Interoperability Working Group (DDRI) and the Metadata Standards Directory Working Group (MSD) are confronting with these issues. In particular, MSD is promoting the building of a directory of discipline-specific metadata standards [13].

Scientific data repositories should *(a)* cooperate with the research community to identify best practices for the production of appropriate documentation, *(b)* contribute to the dissemination of these best practices, e.g., by giving visibility and advice through their web sites, *(c)* offer facilities for supporting the production of documentation compliant with the envisaged best practices, e.g., automatic generation of the metadata, *(d)* put in place pre-publication validation checks aiming at controlling that whenever additional documentation is produced it matches the goals, and, *(e)* encourage and promote the production of third party documentation resulting from real use experiences via the post-publication validation.

Dataset Formatting

The willingness of scientific data repositories to manage any research data has actually pros and cons. Among the cons there is the fact that selected repositories can make no assumptions on the data typologies they are requested to manage. This leads to the development of approaches that tend to be generic and open. Although properly dealing with formats is a pre-requisite for making use of the published data, there is a lack of shared understanding of what a data format must be.

The Research Data Alliance (RDA) Data Type Registries Working Group [25] was created to discuss the issues related with making it possible to associate an intelligible type to a dataset. The notion of type promoted in this WG is aiming at characterising the dataset at multiple levels of granularity, i.e., from individual data points up to the entire dataset. Moreover, WG members expect that types are standardised, unique and discoverable. Finally, they propose to use a registry-based approach to make it possible for data consumers to discover an accurate description of a given type as well as any additional information for managing such a type, e.g., potential software for processing data of the given type. However, the potential limitations of such an approach have been discussed in [71].

Scientific data repositories should encourage data publishers to make the data they are willing to publish available in formats permitting their cross disciplinary use. As a consequence they should have the duty, in cooperation with the rest of actors involved in the data publishing endeavour, to identify and promote data formats that are intelligible as much as possible. The use of standards is a good practice. This can be achieved *(a)* at submission time, i.e., repositories advice the publisher on the potential limitations resulting from the upload of a dataset in a certain format, and *(b)* at access time, i.e., repositories enlarge the set of formats a dataset can be accessed by highlighting whether the format is natively provided or automatically generated by the repository. The set of formats a dataset is exposed at access time might be either a pre-defined list or the result of a negotiation among the data consumer and the registry aiming at identifying the format that fits with the purpose of the consumer.

Dataset Availability

The almost open ended set of dataset typologies expected to be managed by the selected repositories makes the attempts to guarantee present and future availability quite challenging. There exist international standards, models, and best practices for long-term preservation but they are no more able to cover all dataset typologies while it is fundamentally important that preservation policy be structured according to the different goals, priorities, and capabilities of each organisation.

The Dryad data preservation policy is discussed in [67]. The paper states that Dryad's policy development was aided significantly by a review of existing data preservation policies, although the Dryad staff were unable to identify a policy that could be immediately adopted. Dryad is making efforts to adopt preservation standards, models, and best practices. However, understanding that some departure from existing standards is necessary, the policy is continually revised and re-evaluated as preservation practices are refined. Thus the development of Dryad's preservation policy is presented as just the first step in an ongoing process of preservation efforts., i.e., as a living document that should evolve alongside the repository. Hopefully, Dryad's policy has the potential to inform preservation policy development at other repositories.

A detailed discussion on access and use control policies used by repositories is given by Eschenfelder and Johnson [46]. Their analysis explores the subset of repositories they call "controlled data collection", i.e., repositories where staff, or use communities, make and enforce rules to control who can access data or how data can be used.

Dataset Discovery

Independently of the effectiveness of the well known proposed approaches for supporting the discovery of the dataset, it is evident that repositories do not support any proactive facility for alerting users about the availability of datasets. Further, the set of current user-driven data discovery facilities is borrowed from existing domains with very limited adaptations to the data case. They are essentially based on the content of the metadata associated with the datasets, thus the effectiveness of the discovery facilities heavily depends on the quality of such metadata. There is no facility, for end users to enrich the available metadata, e.g., with tags and ratings, thus there is no possibility for end users to rely on this information at discovery time.

Data discovery is among the key facilities a repository should offer on its own content. The facility is expected to be offered by the single repository as well as by any third party service provider willing to build an unifying information space of the existing datasets across existing repositories. This feature is largely relying on metadata, thus the quality of metadata severely impact on the implementation of the facility, e.g., [87]. Besides the need to enhance the quality of metadata, it is fundamental to enlarge both *(a)* the quantity of dataset documentation the facility relies on to build the index and *(b)* the offering of data discovery facilities by envisaging new modes capable to serve also expert users locking for precision in results.

Repositories should invest in reinforcing their offering towards three main directions: *(i)* strengthening the quality of the documentation they collect and associate to the datasets; *(ii)* enlarging the quantity of documentation they collect and associate to the datasets; and *(iii)* empowering the set of discovery facilities offered. On how to improve the quality and quality of the documentation associated to the dataset, we have discussed in Sec. 3.1, e.g., the existence of data papers contributes to dataset discovery. Further discussions and proposals are in the dataset validation part (cf. Sec. 2.1.2). On how to empower the set of data discovery facility, it is just a matter of analysing the plethora of existing facilities aiming at making it possible for users to "discover resources" on the web and apply them to the datasets case. These facilities goes well beyond the case of Google-like search engines. For instance, it is possible to envisage recommender systems for datasets [20].

Dataset Citation

All of these approaches match, to some extent, with the 8 basic principles characterising good citation practices recently developed by the Force11 organisation [51][23], i.e., importance, credit and attribution, evidence, unique identification, access, persistence, specificity and verifiability, interoperability and flexibility. For *Importance*, repositories definitely consider data as first class entities. For *Credit and Attribution*, repositories contribute to give credit to data contributors. However, they have not developed any micro attribution oriented facility aiming at highlighting who did what [2]. For *Evidence*, repositories do support the processes of scholarly authors when need to cite data their claims are based on. However, there is no guarantee that this will happen. For *Unique Identification*, repositories rely on DOIs a mechanism that is "machine actionable, globally unique", and, to some extent, "widely used". However, the limitation of this approach when dealing with subset of datasets are known [38]. For *Access*, the mechanisms in place do not necessarily guarantee a facilitated access to code and other materials enabling "both humans and machines to make use of the reference data". It is up to the data submitter to properly use the metadata and documentation facilities offered by the repositories to try to convey this information also. For *Persistence*, repositories guarantee that the datasets and their associated documentation are and will be available. For *Specificity and Variability*, the mechanisms offered allows to refer an entire dataset only. Nothing is offered to "facilitate verifying that the specific time slice, version and/or granular portion of data retrieved subsequently is the same as was originally cited". For the *Interoperability and Flexibility*, it is a matter of tradeoff between the need to not differ so much and the need to accommodate the variant practices among communities. Actually, every repository offer its own method that is independent from the community and does not offer any facility for a community to customise the way its own data should be cited.

Repositories should support human users willing to cite a dataset to actually cite one of the data papers associated with it. Moreover, they should contribute to the initiatives leading to machine readable citations and implement mechanisms compliant with de facto standards once agreed.

Dataset Validation

Discussion about quality of data in scientific data repositories called to manage any data is at its initial stage. It is characterised by attempts to define what "quality" is and what criteria are to be applied in reviewing as well as in validation of data. Why validation practices are so scarcely applied by data repositories is discussed in a recent blog post [59]. Among the suggestion, is to "forget quality, consider fitness for purpose". The motivation for this is that a dataset may be good enough for one purpose but not another. "Trying to assess the general "quality" of a dataset is hopeless; consider instead whether the dataset is suited to a particular use. Extending the previous idea, documentation of how and in what contexts a dataset has been used may be more informative than an assessment of abstract quality".

We believe that the dataset validation is *(a)* a shared responsibility, there is no single actor in the scientific data publishing scenario that can take the responsibility of assessing the validity of the dataset; *(b)* a continuous procedure, published datasets are not confined to be used by a given community or a use case only thus any attempt to use a published dataset potentially leads to a validation assessment; *(c)* a many facets question, it has to do with technical, scientific, and organisational aspects (to cite a few) all requiring diverse domain expertise; and, *(d)* not a matter of certification, it is almost impossible to envisage a certification scheme that guarantee that the datasets published by a repository are "valid".

Repository should put in place processes to support both pre and post publication validation although with different levels of engagement. Repositories should implement pre-publication validation procedures and resources aiming at assessing the published artifact (both the dataset and the documentation) by using the "average consumer" perspective. This perspective consists in analysing the artifact without any domain specific knowledge about the dataset. It aims at guaranteeing that the data are accompanied with enough documentation to assess whether the dataset fit for his/her purpose. Thus, it is a duty of the repository to guarantee that the artifacts it publishes are "valid" for an average consumer. The procedure should be clearly defined and known to the users of the repository service. Repositories should also support post-publication validation by providing mechanisms for real users to give concrete and documented feedback to both the repository and the data provider resulting from attempts to reuse the dataset. This feedback must have the traits of a scientific publication, actually it should be a research outcome worth to enter in the scholarly record. The authors must give arguments on their experience in using the dataset including the scientific questions they are posing, the application domain, the competing interests. This feedback importantly contribute to form the documentation accompanying a published dataset. In addition to this explicit and feature-rich feedback, the advances resulting from the altmetrics [1] research aiming at collecting diverse "flavours" of impact of scholarly outputs including datasets should be heavily exploited by repositories [53][4].

Dataset Publication Costs

Dryad is the only repository in our sample that always requires submitters to pay a charge independently of the files size. However, the other repositories that offer free submission certainly have relevant costs and one might wonder how they will continue to support them. A partial answer is that CSIRO, Zenodo and 3TU all are institutional repositories and, as such, they are financed by the institutions that have founded them as a step towards their mission. As for Figshare, it is qualified as a non-profit organisation in the re3data.org Registry of Research Data Repositories, and officially presented as a “product“ of the company Digital Science. Figshare offers free depositing and downloading. Recently, however, its policies seem to deal with cost issues by offering optional review processes and implementing “institutional customers” provided with curation workflow functionality allowing administrators to add additional metadata, do any review of the digital content and apply restricted access or embargoes where relevant. Further, it has started a partnership with Taylor & Francis Publishers to offer access to material supplemental to research articles published in Taylor & Francis’s journals [42], and is concluding a partnership with Loughborough University, Symplectic and Arkivum to create UK’s first integrated research data management solution.

Cost issues are already emerging as strictly related with other issue regarding the key role of data repositories in the scientific research. Possibly the separation of scientific repositories into two different worlds characterised as being publicly or non-publicly financed will not hold. As science repositories become greater and greater, public institutions will possibly become not capable of supporting fully-managed and secure service for long-term data retention with online access and a guarantee of data integrity. On the other side, private companies will more and more have need to access and exploit state-of-the-art research in digital preservation that helps them meet their commitments to their customers.

Repositories should liaise with the rest actors involved in the data publishing arena to keep the costs fairly distributed.

2.1.3 Enhanced Publications

Besides the data publishing phenomena, some initiatives in the Research Publishing domain are studying the problem of enhancing the traditional publications. The aim is twofold: *(i)* to pursue an approach characterised by comprehension of the parts interconnected leading to a traditional paper, where publishing includes any product (e.g. publications, datasets, experiments, web services, presentations) and *(ii)* to offer researchers all the elements to repeat (“same experiment, same lab”), replicate (“same experiment, different lab”), reproduce (“same experiment, different configuration”), or reuse (“include part of the experiment into another experiment”) any research activity experiment, here intended as the methodological processes or ICT-based workflows necessary to achieve given scientific conclusions.

In the following, we shall present various definitions and models for Enhanced Publications found in literature, accompanied by the enabling technologies (Enhanced Publication Information Systems) supporting them and the relative benefits in terms of publishing research results together with their experimental context.

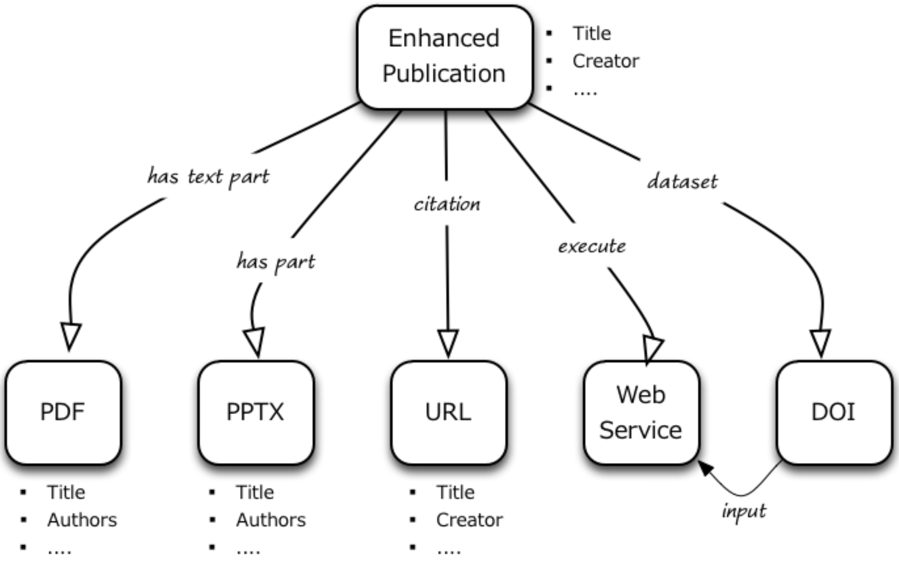


Figure 2.3: Enhanced Publication Example

Enhanced Publication Understandings

Bardi and Manghi from ISTI-CNR [15] defines Enhanced Publication (EP) as in the following:

> *"Enhanced publications are digital objects characterised by an identifier (possibly a persistent identifier) and by descriptive metadata information. The constituent components of an enhanced publication include one mandatory textual narration part (the description of the research) and a set of interconnected sub-parts. Parts may have or not have an identifier and relative metadata descriptions and are connected by semantic relationships".*

From this definition it clearly comes out that Enhanced Publication must include one mandatory textual narration part. Moreover, they provide access to other interconnected parts e.g., datasets, images, tables, workflows, devices, services where each part is accompanied by metadata description and by semantic relationships. This first definition clarifies the scope of this type of digital publication and identifies the technological challenges in supporting it. Fig. 2.3 shows an Enhanced Publication Example.

The SURF foundation [91], which unites Dutch research universities, universities of applied sciences, and research institutions, defines the concept of Enhanced Publication as:

> *"An enhanced publication is a modern, digital format for publishing research data. Such publications are enhanced with research data, graphics, models and so on. They also provide meaningful explanations about how different research results fit together, thus giving other researchers a single, comprehensive overview of all relevant knowledge".*

The OpenAIRE project [77], the Open Access Infrastructure for Research in Europe funded by the European Commission in FP7, which is also the mandatory repository for any peer reviewed journal article belonging to EU projects receiving HORIZON 2020 funding, defines Enhanced Publication as:

> *"An enhanced publication (EP) is a totally new way of publishing in which a traditional publication (a book, an article or a report) is enriched with additional information. An enhanced publication relies on the linking possibilities of the web".*

In these two cases the definition is quite vague and deliberately so. We think this unclearness reflects the fact that the concept of a Enhanced Publication is indeed evolving and being too prescriptive in the definition might prevent innovation. However, it is worth noting that the accent is given to the fact that an EP facilitate researchers in understanding how different research results fit together and that the constituent parts of EPs rely on the linking possibilities of the web.

Finally, an alternative definition of EP is given by Bechhofer at al. [18] from University of Manchester and Oxford under a different name: “Research Object” they define them as:

> *“a semantically rich aggregations of resources that bring together data, methods and people in scientific investigations. Their goal is to create a class of artifacts that can encapsulate our digital knowledge and provide a mechanism for sharing and discovering assets of reusable research and scientific knowledge*”

A Research Object illustration, which well exemplifies their scope, taken from the researchobject.org website[2], is shown in Figure 2.4. Even in this case the traditional

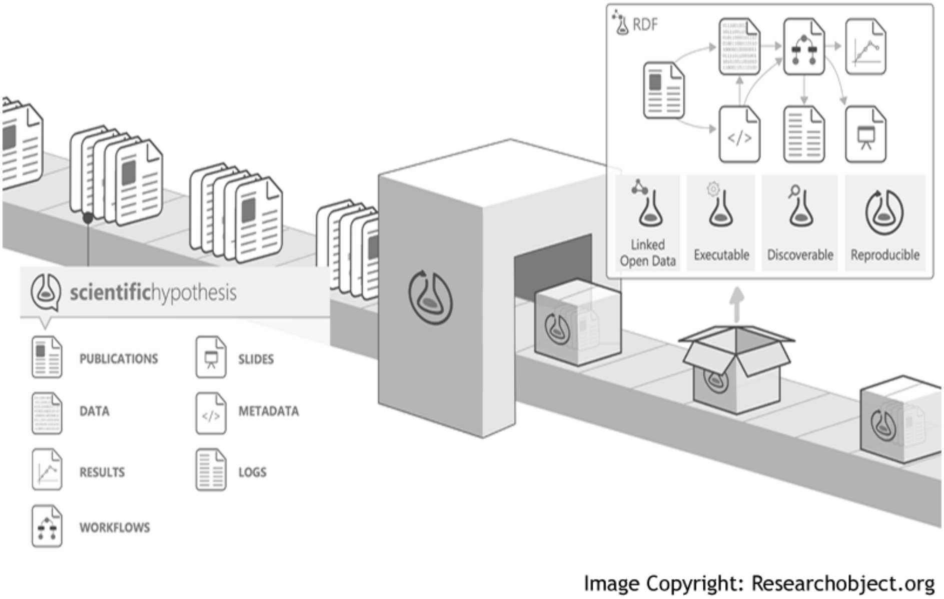


Figure 2.4: A Research Object illustration

publication is enriched with research data, graphics, workflows and so on.

[2] http://www.researchobject.org

Data Models for Enhanced Publications

As previously stated the parts of an enhanced publication must include one textual narration part (the description of the research work) and a set of interconnected constituents. These constituents could either have or not a unique identifier where each of these is accompanied by metadata description and by semantic relationships. Data Models for EPs define the organisation structure of these constituents in different manners. In some approaches the constituents are referenced, shared or passed as inputs to workflow engines, in others instead they are grouped in "Packages" embedding all of them. Bardi et al. [14] studied EP data models, identifying recurrence schemes of *(i)* the constituent parts, *(ii)* the associated metadata and *(iii)* interrelations among them, and conceived the following types of enhanced publication parts:

- *Embedded parts:* parts that include supplementary material files;
- *Structured-text parts:* parts providing an editorial structure of its textual subcomponents;
- *Reference parts*: parts that include URLs to external objects;
- *Executable parts*: parts that include software and data to run an experiment;
- *Generated parts:* parts that include dynamically parts that may change depending on updates of given input research data.

Enabling Technologies

Information systems for enhanced publications (EPIS) increase in number year by year in literature, they aim at providing scientists with a range of functionalities handling enhanced publications complying with heterogeneous data models. Further, they offer management and consumption functionalities for enhanced publications, the former for the creation, updates and deletion, the latter for their exploitation e.g., reading, executing, sharing etc.

EPIS in literature are surveyed, clearly classified and discussed by Bardi et al. [14], according to them EPIS can be clustered by *functionality goals*, these are:

- packaging publications with other research material;
- improving reading experience and understanding via thin clients (browser or desktop applications);
- creating relationships between publications and research datasets;
- enabling repetition of scientific experiments.

However, we found very few EPIS available in the research marketplace and actually exploitable, among these we picked two EPIS deserving an in-depth analysis in our opinion: *Utopia Documents* [11], funded by the Open PHACTS consortium[3], which consists of a downloadable desktop application comprising a PDF reader that integrates visualisation and data-analysis tools with published research articles, and *MyExperiment* [41], co-funded by JISC[4], EPSRC, and Microsoft, whose purpose is to enable scientists to create, share, re-use and re-purpose workflows for data analysis.

In the following, we present the functionalities these two systems provide to support research publishing, and analyse them from their capacity of fulfilling the functionality goals reported in the previous.

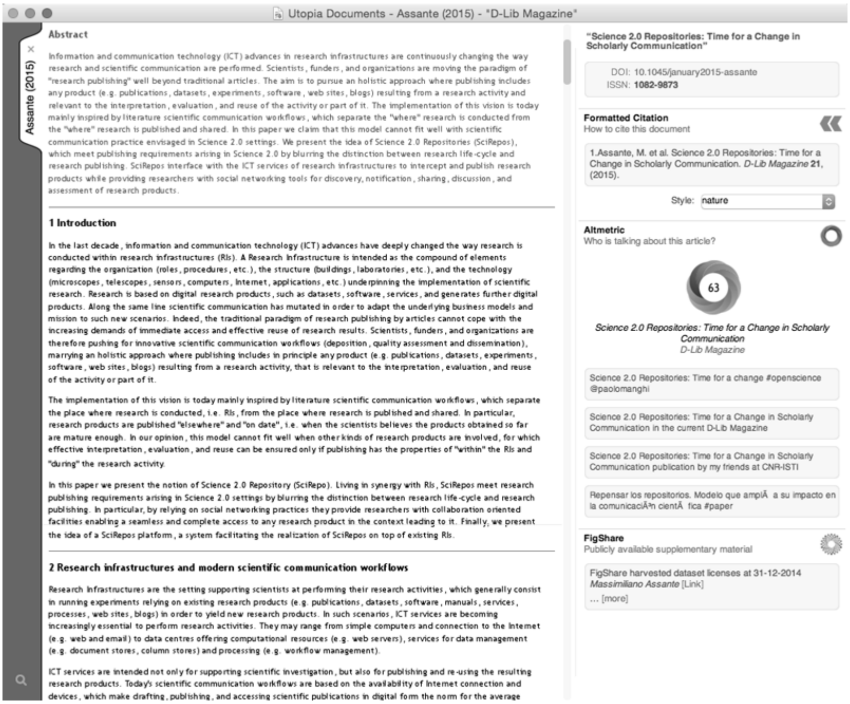
Utopia Documents

The *Utopia Documents* desktop application is released under an Open Source license and provide functions working particularly well with life science articles. The software is capable provide users with dynamic views of tables and images (within the Life Science domain) and to show a set of information and data resources relating to the article. Specifically, when opening an article (pdf), the application analyses its content, and tries to connect it to the online data resources available in the following data sources:

- *FigShare*: to provide direct links to the research data connected to the publication. Please note that FigShare is one of the Scientific Data Repositories we surveyed in the previous section.
- *Altmetric*: one of the internet available services dealing with altmetrics [1], i.e., metrics aiming at measuring the impact of research products [82]. The Altmetric service scans social networking sites such as Twitter, Facebook etc., online blogs, etc. looking for mentions of scholarly articles and generates a score based on the quality and quantity of attention they received.
- *Mendeley Web*: Mendeley Web [54] is an online social network for researchers providing facilities to discover the latest research in a given domain.
- *CrossRef:* CrossRef is an association of scholarly publishers that develops shared infrastructure to support more effective scholarly communications: According to their website (crossref.org), their citation-linking network today covers over 72 million journal articles and other content items (books chapters, data, theses, technical reports).
- *Sherpa/RoMEO*: Sherpa/RoMEO provides information about the article's publisher copyright and archiving policies.

[3] Open PHACTS consortium: http://www.openphacts.org/about-open-phacts/about-open-phacts

[4] JISC: the United Kingdom committee, which inspires UK colleges and universities in the innovative use of digital technologies.



Abstract

Information and communication technology (ICT) advances in research infrastructures are continuously changing the way research and scientific communication are performed. Scientists, funders, and organizations are moving the paradigm of "research publishing" well beyond traditional articles. The aim is to pursue an holistic approach where publishing includes any product (e.g. publications, datasets, experiments, software, web sites, blogs) resulting from a research activity and relevant to the interpretation, evaluation, and reuse of the activity or part of it. The implementation of this vision is today mainly inspired by literature scientific communication workflows, which separate the "where" research is conducted from the "where" research is published and shared. In this paper we claim that this model cannot fit well with scientific communication practice envisaged in Science 2.0 settings. We present the idea of Science 2.0 Repositories (SciRepos), which meet publishing requirements arising in Science 2.0 by blurring the distinction between research life-cycle and research publishing. SciRepos interface with the ICT services of research infrastructures to intercept and publish research products while providing researchers with social networking tools for discovery, notification, sharing, discussion, and assessment of research products.

1 Introduction

In the last decade, information and communication technology (ICT) advances have deeply changed the way research is conducted within research infrastructures (RIs). A Research Infrastructure is intended as the compound of elements regarding the organization (roles, procedures, etc.), the structure (buildings, laboratories, etc.), and the technology (microscopes, telescopes, sensors, computers, Internet, applications, etc.) underpinning the implementation of scientific research. Research is based on digital research products, such as datasets, software, services, and generates further digital products. Along the same line scientific communication has mutated in order to adapt the underlying business models and mission to such new scenarios. Indeed, the traditional paradigm of research publishing by articles cannot cope with the increasing demands of immediate access and effective reuse of research results. Scientists, funders, and organizations are therefore pushing for innovative scientific communication workflows (deposition, quality assessment and dissemination), marrying an holistic approach where publishing includes in principle any product (e.g. publications, datasets, experiments, software, web sites, blogs) resulting from a research activity, that is relevant to the interpretation, evaluation, and reuse of the activity or part of it.

The implementation of this vision is today mainly inspired by literature scientific communication workflows, which separate the place where research is conducted, i.e. RIs, from the place where research is published and shared. In particular, research products are published "elsewhere" and "on date", i.e. when the scientists believes the products obtained so far are mature enough. In our opinion, this model cannot fit well when other kinds of research products are involved, for which effective interpretation, evaluation, and reuse can be ensured only if publishing has the properties of "within" the RIs and "during" the research activity.

In this paper we present the notion of Science 2.0 Repository (SciRepo). Living in synergy with RIs, SciRepos meet research publishing requirements arising in Science 2.0 settings by blurring the distinction between research life-cycle and research publishing. In particular, by relying on social networking practices they provide researchers with collaboration oriented facilities enabling a seamless and complete access to any research product in the context leading to it. Finally, we present the idea of a SciRepos platform, a system facilitating the realization of SciRepos on top of existing RIs.

2 Research infrastructures and modern scientific communication workflows

Research Infrastructures are the setting supporting scientists at performing their research activities, which generally consist in running experiments relying on existing research products (e.g. publications, datasets, software, manuals, services, processes, web sites, blogs) in order to yield new research products. In such scenarios, ICT services are becoming increasingly essential to perform research activities. They may range from simple computers and connection to the Internet (e.g. web and email) to data centres offering computational resources (e.g. web servers), services for data management (e.g. document stores, column stores) and processing (e.g. workflow management).

ICT services are intended not only for supporting scientific investigation, but also for publishing and re-using the resulting research products. Today's scientific communication workflows are based on the availability of Internet connection and devices, which make drafting, publishing, and accessing scientific publications in digital form the norm for the average

"Science 2.0 Repositories: Time for a Change in Scholarly Communication"

DOI: 10.1045/january2015-assante
ISSN: **1082-9873**

Formatted Citation
How to cite this document

1.Assante, M. et al. Science 2.0 Repositories: Time for a Change in Scholarly Communication. *D-Lib Magazine* **21**, (2015).

Style: nature

Altmetric
Who is talking about this article?

63

Science 2.0 Repositories: Time for a Change in Scholarly Communication
D-Lib Magazine

Science 2.0 Repositories: Time for a change #openscience @paolomanghi

Science 2.0 Repositories: Time for a Change in Scholarly Communication in the current D-Lib Magazine

Science 2.0 Repositories: Time for a Change in Scholarly Communication publication by my friends at CNR-ISTI

Repensar los repositorios. Modelo que amplÃ a su impacto en la comunicaciÃªn cientÃ fica #paper

FigShare
Publicly available supplementary material

FigShare harvested dataset licenses at 31-12-2014
Massimiliano Assante [Link]
... [more]

Figure 2.5: Example of an article opened in Utopia Documents

Apart from the data sources listed above, there are a set of Life Science domain specific sources too such as The Semantic Biochemical Journal, SciBite, Royal Society of Chemistry etc.

Figure 2.5 shows an example of an article opened in the Utopia Documents. Specifically, on the reader's left the article is shown while the sidebar on the right shows information relating either to the whole article. The sidebar content is divided into different sections, in the example we can see, starting from the top: *(i)* a formatted citation for the article, provided by CrossRef, *(ii)* the Altmetric score for the article, and *(iii)* links to article's dataset published on FigShare.

Overall, Utopia Documents is an EPIS supporting two out of four of the functionality goals described in the previous (improving reading experience and understanding and creating relationships between publications and research datasets) and its data model is comprised of: *Embedded parts*, *Structured-text parts, Reference parts*, and (partially) *Generated parts*.

MyExperiment

According to the MyExperiment website[5], MyExperiment is:

> *“a workflow repository and a social networking site for scientists, but instead of music or photos scientist are encouraged to share scientific data and know-how*”.

The website also reports that:

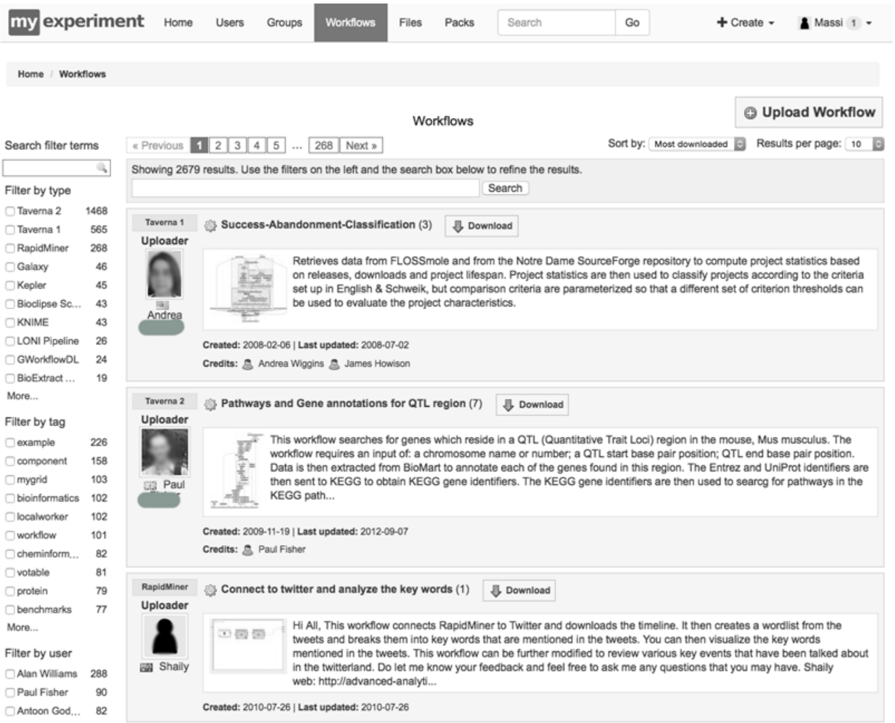
> *“unlike Facebook or MySpace, myExperiment fully understands the needs of the researcher and makes it really easy for the next generation of scientists to contribute to a pool of scientific methods, build communities and form relationships — reducing time-to-experiment, sharing expertise and avoiding reinvention*”.

In our opinion, MyExperiment basically focuses on providing users with mechanisms to support the sharing of workflows. The types of workflows supported (written in a range of workflow languages) are the following: Taverna, Galaxy, Rapid Miner, Bio Extract, and Kepler. Apart from workflow you can share “packs”. Packs group together related workflows and files, allowing users to download the whole set of the EP’s constituents, which eventually consists of a zip file.

Users are the core dimension for myExperiment. They can be clustered in: *(i)* developers interested in contributing their workflows into the repository for subsequent sharing with the scientific community, and *(ii)* scientists wishing to discover workflows to be reused in their own research. Although MyExperiment content can be freely consulted and downloaded by anonymous users, a richer user experience is available when they register to the system. Once registered, users can exploit two mechanisms to form communities. The former, a user may ask for friendship from other registered users in order to build his/her network of trusted peers. The latter, a user can set up a group for which he/she is automatically set as administrator and, successively, can invite other users to enter in the group, or other users may ask for being added in a group.

Figure 2.6 depicts a screenshot of the MyExperiment workflows web page showing the most downloaded workflows submitted to myExperiment. The *discovery* of workflows in myExperiment can be performed by using a Search functionality, and by using the filtering mechanisms placed on the sidebar. It is possible to filter by type, tag, and user. The *sharing* of workflows can be performed throughout the social infrastructure provided by myExperiment around its workflow repository. Users can upload their workflow files in their native format and have possibilities to add a title, a

[5] MyExperiment website: http://www.myexperiment.org/



Workflows

Search filter terms

Sort by: Most downloaded

Results per page: 10

Showing 2679 results. Use the filters on the left and the search box below to refine the results.

Filter by type

□ Taverna 2	1468
□ Taverna 1	565
□ RapidMiner	268
□ Galaxy	46
□ Kepler	45
□ Bioclipse Sc…	43
□ KNIME	43
□ LONI Pipeline	26
□ GWorkflowDL	24
□ BioExtract …	19

More...

Filter by tag

□ example	226
□ component	158
□ mygrid	103
□ bioinformatics	102
□ localworker	102
□ workflow	101
□ cheminform…	82
□ votable	81
□ protein	79
□ benchmarks	77

More...

Filter by user

□ Alan Williams	288
□ Paul Fisher	90
□ Antoon God…	82

Taverna 1

Success-Abandonment-Classification (3)

Andrea [redacted]

Retrieves data from FLOSSmole and from the Notre Dame SourceForge repository to compute project statistics based on releases, downloads and project lifespan. Project statistics are then used to classify projects according to the criteria set up in English & Schweik, but comparison criteria are parameterized so that a different set of criterion thresholds can be used to evaluate the project characteristics.

Created: 2008-02-06 | **Last updated:** 2008-07-02

Credits: Andrea Wiggins James Howison

Taverna 2

Pathways and Gene annotations for QTL region (7)

Paul [redacted]

This workflow searches for genes which reside in a QTL (Quantitative Trait Loci) region in the mouse, Mus musculus. The workflow requires an input of: a chromosome name or number; a QTL start base pair position; QTL end base pair position. Data is then extracted from BioMart to annotate each of the genes found in this region. The Entrez and UniProt identifiers are then sent to KEGG to obtain KEGG gene identifiers. The KEGG gene identifiers are then used to searcg for pathways in the KEGG path...

Created: 2009-11-19 | **Last updated:** 2012-09-07

Credits: Paul Fisher

RapidMiner

Connect to twitter and analyze the key words (1)

Shaily

Hi All, This workflow connects RapidMiner to Twitter and downloads the timeline. It then creates a wordlist from the tweets and breaks them into key words that are mentioned in the tweets. You can then visualize the key words mentioned in the tweets. This workflow can be further modified to review various key events that have been talked about in the twitterland. Do let me know your feedback and feel free to ask me any questions that you may have. Shaily web: http://advanced-analyti...

Created: 2010-07-26 | **Last updated:** 2010-07-26

Figure 2.6: A screenshot of the MyExperiment workflows web page

description, and keywords. It is also possible to associate workflows with citations for interlinking with a publication.

Overall, myExperiment is a general repository for “multi-language” workflows and related objects. Its focus is to enable sharing and reuse of digital experimental protocols and support reproducible science. It supports two out of four of the EPIS functionality goals described in the previous, even though the scientific workflow is its focus, not the narrative part. Specifically, *(i)* packaging publications with other research material can be supported but it is not mandatory (*ii*) enabling repetition of scientific experiments is fully supported. Finally, creating relationships between publications and research datasets is not supported, but it is possible to create relationships between publications and workflows. As a consequence, the Research Object data model is comprised of: embedded, reference, and executable parts.

2.2 Key Findings

The *Data Publishing* and *Enhanced Publications* initiatives reported in this chapter demonstrate and attest the fact that researchers, scientists or experts in a field are moving towards a newer interpretation of Research Publishing characterised by the belief that the parts and the processes comprising and leading to a scientific achievement, are thoroughly interconnected and explicable only by reference to the whole. These actors demand new services capable of publishing literature, datasets, experiments, any form of research outcome perceived to be important for the interpretation and reuse of scientific results. After all, nowadays the continuous ICT services technological advances in terms of volume, velocity and variety of data they can manage and processing power they can use can facilitate this transition towards *modern scientific communication workflows.* ICT services are certainly capable of supporting and make this transition possible.

The resulting benefits are reported in the following:

- better interpretation of scientific results;
- more rigorous, possibly automated, evaluation of the research outcomes;
- omni comprehensive scientific reward practices;
- maximisation of research reuse, thereby reducing the costs of research.

We have seen that an important step towards Research Publishing in Science 2.0 and the establishment of modern scientific communication workflows is certainly carried out by information systems falling under the Enhanced Publication umbrella, which we presented and analysed in 2.1.3. During our investigation we found a large number of scientific literature papers proposing EPIS, since 2007 at least. However, so far few of them have exited from a prototype stage and have been adopted by a considerable number of researchers or stable community.

We feel that the reason behind the slow adoption of EPIS, and more generally the reason behind the slow growth and establishment of modern scientific communication workflows is methodological. In fact, most practices and technologies proposed today to renovate scientific communication tend to reflect the literature publishing workflows. Research activities are conducted within Research Infrastructures, while publishing of the relative publications takes place “elsewhere”, on the research marketplace, and “on date”, when the researcher believes the publication is mature enough (cf. Fig. 2.1). This fact impedes, and in some cases discourages, scientists willing to share their results because publication and interlinking of research data happens too often *a posteriori* and requires additional work. The same methodology and attitude is applied to research data publishing. As a matter of fact, any traditional publication related product, such as its embedded or reference part, has to be published into scientific data repositories or into data journals (cf. Sec 2.1.1), or into experiment/workflow repositories like myExperiment (cf. Sec 2.1.3).

Apart from these methodological reasons, publishing research products different from scientific literature is hindered by several “cultural barriers” [21] [24], intended as the partial or full absence of well-established communication workflows defining what it means to publish, peer-review, citing, and guarantee scientific reward for products that are different from the traditional scientific publication [81]. Such lack of common understanding results in the fact that research products of several kinds remain in “the researchers desk drawer” or in the ICT services of RIs.

Therefore, the idea behind this approach, is that of simply extending the marketplace with new sources dedicated to publish new kinds of digital products, and to implement submission and peer-review tools similar to those existing for literature.

2.2.1 Drawbacks affecting Publishing Practices

Although we reckon this approach to be pragmatically correct, since researchers are accustomed to this way of thinking, we also believe that its immediate side-effects are counter productive and hindering the implementation of proper scientific communication in Science 2.0. This is because Science 2.0 research activity, being supported by RI-oriented ICT services, is *(i)* strongly contextualised and *(ii)* intrinsically dynamic; these features and requirements conflict with the “elsewhere” and “on date” philosophy of literature scientific communication workflows. According to these, products leave the RI ICT services to be transferred and deposited in marketplace repositories of specific kind (e.g., repositories for publications, datasets, and experiments). As such they are subject to the relative metadata/file deposition idiosyncrasies and management policies.

As a consequence, published products suffer from the potential problems usually associated with scientific communication, i.e., *no communication*, *slow communication, incomplete communication*, *inaccurate communication*, or *unmodifiable communication* [75] and from some *drawbacks*, embracing the 3 publishing phases described at the beginning of this Chapter (Deposition, Quality Assessment, Dissemination) that we discuss in the following:

Drawbacks during Deposition

- **Decontextualisation**: Although once published research products are annotated with metadata referring to the context leading to them (e.g., provenance), in the reality these products are deprived of any relationship to the original research activity, i.e., the notion of research activity does not survive in an effective way in marketplace repositories;

- **Staticity:** Published products are frozen to their publishing status, i.e. marketplace repositories often contain snapshots at the time of publishing of the products and they are not concerned with their evolution over time;
- **Extra Cost:** Expensive to transfer and maintain when copies of the products need preparation before being transferred (e.g., anonymisation) or entail hardware and administration cost for their management (e.g., disks, synchronisation, IPR issues).

Drawbacks during Quality Assessment

- **Ineffective peer-review:** The real evaluation of research products other than papers can be hardly performed out of the scope of the research activity or RI and in most cases without the support of dedicated ICT services (e.g., evaluate quality of large datasets or alternative products).

Drawbacks during Dissemination

- **Fragmentation:** Research products are scattered across several marketplace repositories, i.e. scientists willing to re-use products published by others must interact with several end-points to find what they want (e.g., Google Scholar, DataCite, Google, repositories); for some products (e.g., blogs, websites), such sources are only search engines, since there is no dedicated marketplace repository for their publishing;
- **Lack of semantic linking:** There is no guarantee that published products contain relationships between them, since such links have to be specified and maintained overtime by the authors across several marketplace repositories (e.g., dataset and publication repositories) or are not even maintained by such repositories.

We believe these drawbacks limits the effective interpretation of research results, hence their correct evaluation and reuse, and reduces the number of products eligible for publishing. For example, the staticity issue is important when the RI production chain is characterised by high velocity and dynamicity, it is non-trivial to decide which products and when are worth publishing; e.g., datasets can be dynamic (e.g., versioned, staged, query results), and deciding which stage/version of the data should be published implies some form of selection.

As a consequence, some products may live in their RIs but never be published due to the implicit drawbacks of publishing. The ineffective peer-review issue is important instead when we have to deal with research products other than papers, datasets for instance. In fact, discussion about the quality of dataset in scientific data repositories is at its initial stage. It is characterised by attempts to define what “quality” is and what criteria are to be applied in reviewing as well as in validation of data, this last is referred to any process, human or not, whose purpose is the one of assessing the “quality of being clear” or “soundness” of the published data.

3

Science 2.0 Repositories

In the previous chapter we presented and surveyed the practices and approaches for data publishing promoted by data journals and scientific data repositories and proved that publishing research products other than papers is gaining momentum. We then discussed the current alternatives to traditional publications that can be found in literature and analysed them arguing that they still contain methodological and technical barriers to modern scientific communication. We supported this identifying a set of drawbacks affecting the current publishing practices and by showing that their implementation is mainly inspired by literature scientific communication workflows, which separate the "where" research is conducted from the "where" research is published and shared.

In this chapter we claim that this model cannot fit well with scientific communication practice envisaged in Science 2.0 settings and present the notion of Science 2.0 Repository (SciRepo) as a possible solution. Living in synergy with RIs, SciRepos meet research publishing requirements arising in Science 2.0 settings by blurring the distinction between research life-cycle and research publishing. Specifically, by relying on social networking practices they provide researchers with collaboration oriented facilities enabling a seamless and complete access to any research product in the context leading to it. The chapter continues presenting a Reference Model and concludes reporting on the benefits of research publishing using SciRepo.

3.1 Defining Science 2.0 Repositories

We believe that in order to enable effective scientific communication workflows, research product creation and publishing should both occur “within” the RI (as opposed to “elsewhere”) and “during” the research activities (as opposed to “on date”) (cf. Sec. 2.2.1). To make this possible, research infrastructures ICT services should not only be conceived to provide scientists with facilities for carrying out their research activities, but also to support marketplace like facilities, enabling RI scientists to publish products created by research activities and other scientists to discover and reuse them. In other words, RIs should not rely on third-party marketplace sources to publish their products, rather should integrate them into the RI.

Such a merge between research infrastructure and research marketplace would overcome known “cultural barriers” and the “methodological barriers” mentioned in Section 2.2. In the RI scope, research products publishing would *(i)* be facilitated by the very ICT services that are generating them, *(ii)* take advantage of research activity awareness and product interlinking, *(iii)* support access rights issues that fit the need of the community, and *(iv)* subsume the major costs of storing and curating products. In other words, by bringing marketplace features within the RI, researchers can finally achieve their best effort in terms of scholarly communication.

We surveyed the current repository platforms in 2.1.2 and saw that unfortunately, they are not apt to implement this vision, as they are designed not to integrate with existing RI ICT services but to support instead today’s notion of “elsewhere” and “on date” research marketplace. Therefore, we propose an innovative class of repositories, named Science 2.0 Repositories (SciRepos). SciRepos are characterised by the following features:

- Integrate with RI ICT services in order to intercept the generation of products within research activities and publish such products, that is making them discoverable and accessible to other researchers;
- Provide scientists with repository-like tools for accessing and sharing research products generated during their research activities;
- Rely on social networking practices [43][93] thus to modernise (scientific) communication both intra-RI and inter-RI, e.g., posting rather than deposition, “like” and “open discussions” for quality assessment, sharing rather than dissemination.

Figure 3.1 illustrates a SciRepo running on an hypothetical RI. The RI enables two research activities RA_1 and RA_2 and keeps track of the executed experiments, their

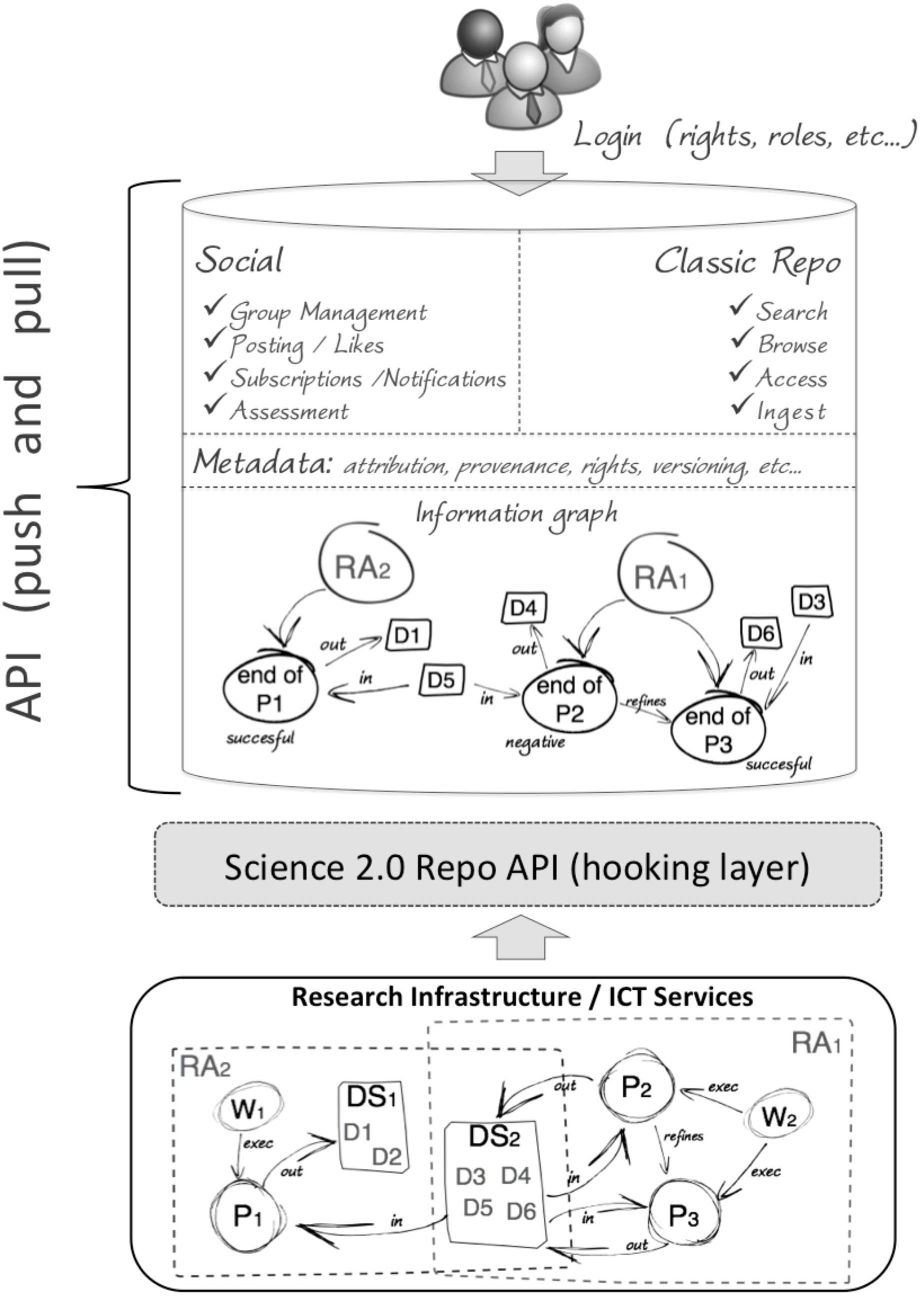


Figure 3.1: Example of SciRepos integrated with Research Infrastructures ICT services

input and output data, and their final status (successful or unsuccessful). For example, in RA_1 researchers run experiments by executing workflow W2. Each execution of the workflow, e.g., processes P2 and P3, collects input data and deposits output data from and into the local store DS2. In particular, process P3 has refined the unsuccessful execution P2 of W2, and improved the experiment to make it successful. The SciRepo sits on top of the RI which interfaces its ICT services with the repository in order to disclose the outcome of research activities to SciRepo users. The figure shows that the repository is provided with a metadata store comprising an information graph useful for representing research activities, their related products, and the relationships between them. Moreover, classic repository facilities make it possible to store product payloads originally residing out of the RI (e.g., publications, alternative products). Products can be of different typologies, e.g., workflows, executed workflows, datasets; their metadata can include different information, e.g., descriptive, attribution, provenance, rights, versioning, execution status, execution parameters, quality; their relationships may represent different associative semantics, e.g., input to process, output to process, refines process. It is important to note that the graph is populated automatically by the hooking layer during the research life-cycle and without scientists being directly involved in the actual action of publishing. The SciRepo supports scientists with two kinds of end-user functionalities:

- **Repository-oriented functionalities:** they offer typical repository functionalities on the information graph such as search and browse allowing search by product typology, but also to navigate from research activities to products and related products. It offers ingestion facilities, allowing scientists to manually or semi-automatically upload “external” products into the repository and associate them to a research activity, thus including them in the information graph.

- **Collaboration-oriented functionalities:** they offer typical social networking functionalities to support collaboration between scientists during any research activity, such as the possibility to subscribe to happenings relative to products and be promptly notified, e.g., the completion of a workflow execution, the generation of datasets obeying to some criteria. As a matter of fact research activity collaborations outside of the scirepo context use too often generic tools that are not well integrated (over emails and shared documents for example). In SciRepo users can reply to posts and, most importantly, can express opinions on the quality of products, e.g., “like” actions or similar. More sophisticated assessment/review functionalities (single/double blind) can be supported, in order to provide more traditional notions of quality. Interestingly, posts are themselves a special typology of products of the research activity and are searchable and browsable in the information graph. Collaboration is indeed crucial during the research process, these collaboration-oriented facilities also support everyone involved in a given research activity to obtain the information they need, and overcome classic collaboration problems where, for instance, senior scientists having a wider vision of the research being conducted, may participate in some meetings or may be involved in some email exchanges and discussions where other researchers may not, with the overall result of making collaboration inefficient.

Thus, SciRepos can be considered as RI-oriented sources in the research marketplace. They offer functionalities allowing scientists to publish products automatically or manually, discover and access products according to their metadata descriptions and end-user access rights, peer-review products according to several evaluation models. In addition, they can also integrate interoperability mechanisms to move products in and out the boundaries of the RI. In order to implement a SciRepo, RIs should develop their own software, thereby investing in a direction that requires different kinds of skills and dedicated funds. In order to facilitate this process we are designing and developing a SciRepo platform, conceived to support the implementation of SciRepos at minimum development cost for the RIs, as described in the next.

The Science 2.0 Repository platform is compliant with the Reference Model presented in section 3.2 out-of-the-box. RI developers can take advantage from this model to better understand the publishing expectations of their scientists and by implementing what is envisaged in the hooking layer. Developers can also customise their SciRepo by enriching the data model specification with directives regarding how the different functionalities should be instantiated with respect to it. Most importantly, as defined by the model, the platform exposes a set of APIs required by RI developers to write the “hooks” needed to interconnect their ICT services with the platform and enable “during” publishing workflows. In the following we present the platform reference model and the publishing functionalities it offers.

3.2 Reference Model

SciRepos vary in the nature of the products they support, depending on the specific research context. This reference model is an abstract work intended for understanding significant concepts and relationships among the components of a SciRepo. It is not directly coupled to any standard, technology or other concrete implementation detail. Its purpose is the one of providing common semantic that can be used unambiguously across and between different implementations.

In order to appropriately characterise a SciRepo platform, we decided to look at it from the perspectives of the actors that operate with it. These perspectives highlight the needs of the different actors, use the appropriate terms and definitions, and give the perception of the required relationships. The roles taken into account are three:

- **SciRepo End-users:** the actors that exploit the SciRepo functionality for providing, consuming and managing its content. They perceive the SciRepo as a stateful entity which serves their functional needs through the interaction with it. It is worth noticing that the behaviour and the outcomes of the functionalities depend on the state of the SciRepo at the time of the request, where the state meant here is represented by the information space available plus the pool of users authorised to access them. This state is continuously evolving accordingly to the functionality activated by the various users.

- **SciRepo Administrators:** the actors selecting which RI services or applications a SciRepo should support and decide where and how to deploy the SciRepo. They interact on top of the SciRepo hooking layer (See Fig. 3.1) by enabling specific operations and configuration parameters, i.e configure the SciRepo by enriching the data model specification with directives regarding how the different functionalities should be instantiated with respect to it. For example, directives may specify how end-user interfaces should enable discover and browse of the information space, e.g., which product typology and metadata fields should be displayed, browsable, post-able, assessable or can be used to configure export APIs, e.g., protocol, subset of information graph to be exported.

- **RI Application Developers:** The actors in charge of extending the RI ICT services and applications to exploit the SciRepo API. They interact symmetrically to the SciRepo Administrators, on the bottom of the SciRepo hooking layer through its APIs (See Fig. 3.1). Their role is to embed small programs, namely hooks, into existing RI services which react to RI events (e.g., dataset creation, experiment execution) by calling the hooking layer APIs, that in turn, transform these events in meaningful information. The SciRepo API must be designed as a software system typology capable to evolve in accordance to the user requirements and thus able to fulfil the needs arising in any application scenario with the minimum effort as possible.

These first two roles, taken in the order, identify two different perspectives each of which could be an extension of the previous one. The SciRepo End-users, perceiving only what a specific SciRepo platform provides them, deal with concepts and relationships directly visible accessing and using a SciRepo. The SciRepo Administrators select which RI services or applications a SciRepo should support and decide where and how to deploy the SciRepo. As a consequence, they need a “richer” model of the SciRepo Platform. This model must be suitable to represent not only the functional aspects of the supported SciRepos but, also, the components that implement the functionality and the hosting nodes where these components can be deployed.

The RI Application Developers instead require a more complete representation of the SciRepo Platform data model. Specifically, they need to know how the model specifies the relationships and dependencies among the components characterising the information space and how these are related to the end-user functionality to optimally write the hooks needed to interconnect their ICT services with the platform.

In the following we present the most general concepts of the model. As Fig. 3.2 shows, a SciRepo is called to support users by providing them with a set of functionalities for managing its Information Space. Like any other system, all of this is regulated by policies, e.g., who can do what, and it is characterised by an architecture.

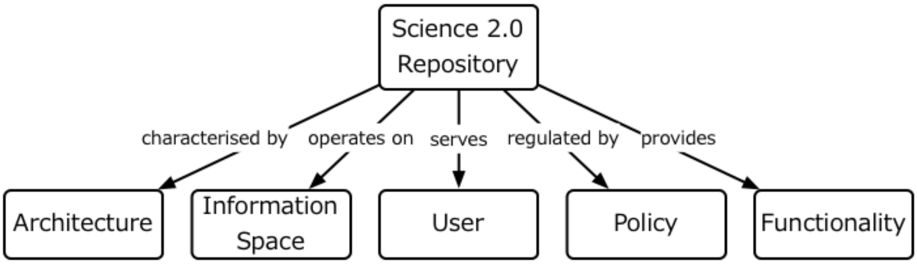


Figure 3.2: The SciRepo Main Concepts

- ***Information Space**:* models the content that is managed by the SciRepo, the content is organised into a classic entity relationship diagram. The diagram contains connected entities (nodes) and relationships capable of contextualising the Research Products in any Research Activity domain.

- ***Functionality***: represents the operations supported by SciRepo. The functionalities a SciRepo is expected to offer are oriented to Science 2.0 settings and to the scholarly communication. The functionalities vary depending on the perspective.

- ***User***: models the actors that exploit the SciRepo. They are consumers of the SciRepo content and/or providers of it (also via the underlying Research Infrastructure services). SciRepo connects its users and support them in performing their research activities by consuming already available RPs to produce new knowledge. Besides, it provides its users with a clear view on what is happening in their Research Activities.

- ***Policy***: regulates the approval to use a functionality on one or more Research Activities in accordance with the role-based access control model explained in the following. Usage rights are modelled as associations between roles, functionality and Research Activities.

- ***Architecture***: models the mapping of functionality onto hardware and software components, the mapping of the software architecture onto the hardware architecture, and human interactions with these components. Unlike the other main four SciRepo concepts, the architecture becomes meaningful and of pertinence of the SciRepo Administrators and RI Application Developers only. As a matter of fact, the SciRepo End-users do not take care of this characteristic because they perceive the SciRepo by interacting with a graphical user interface provided in a web browser. A Reference Architecture for a SciRepo Platform is given in Chapter 4.

For the sake of simplicity, the concepts and relationships of the model are graphically represented through concepts maps [76]. In the rest of this section we present conceptual maps for each of the above identified perspectives.

3.2.1 The SciRepo End-users perspective

The SciRepo end-user perspective is centred around the actions performed by three main actors: the content consumers that access and use the SciRepo; the content providers that can provide content both manually, i.e. by ingesting it or automatically, i.e. by using the RI's ICT services and applications to trigger the creation of new RPs; and the RA managers, whose tasks include describing the RAs in which they are involved together with the management of its users and policies. They perceive the SciRepo Platform through the SciRepo functionality provided by it.

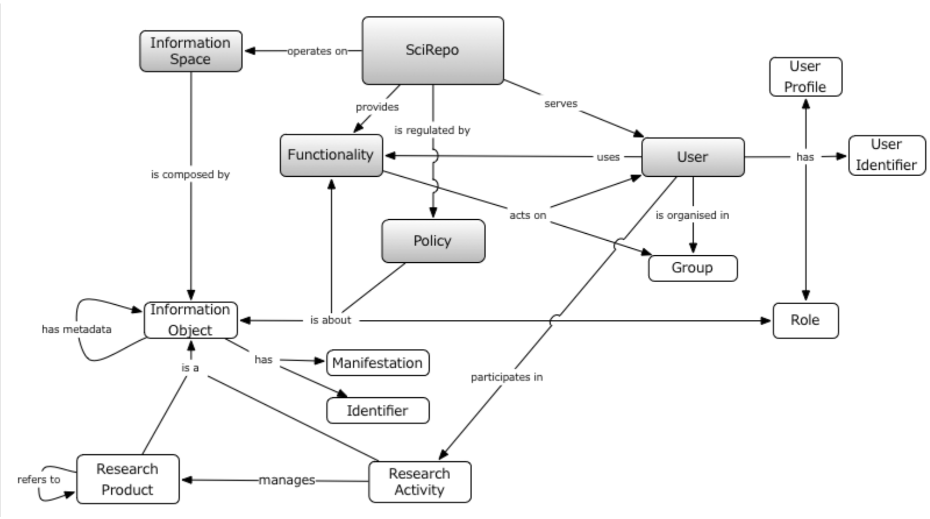


Figure 3.3: The SciRepo End-user Domains Perspective

Figure 3.3 shows the main concepts and relationships of the model that represents the SciRepo from these users' perspective. The *Information Space* and *Functionality* concepts are further detailed with two separate Figures, 3.4 and 3.5 respectively, as these two concepts are the ones most characterising the SciRepo and deserved an in-deep analysis. Consequently, we present the SciRepo concepts and relationships starting from the Information Space and Functionality, following with User and Policy main concepts. Please note that the Architecture concept is missing in this perspective, this is because the SciRepo end-users do not take care of this characteristic as they perceive the SciRepo by interacting with a graphical user interface provided in a web browser.

Information Space

The Information Space perceived by SciRepo end-users is composed by *Information Objects* and, in turn, *Research Activities* (RA) and their related products. Products generated within the scope of a research activity are related with it and can have semantic relationships among them (e.g., citedBy, versionOf, inputDataset). We denominate such products *Research Products* (RP).

Each Information Object has an unique *Identifier*, associated *Metadata* and a *Manifestation*. The *Identifier* represents the minimal information enabling to distinguish one Information Object from all the others. The *Metadata* follows the "classic" definition of metadata, as is "data about data"; it can be used in different contexts with different purposes and is an Information Objects itself; our model captures the needs to have metadata associated to an information object as a mean for enhancing the functionality and in general the management of the object. *Manifestation* is the physical representation of an Information Object; this concept probably is the most important one as regards how the users are provided with the information they are looking for. It is worth noting that we are dealing with digital objects and thus the manifestation is itself a digital object. Examples of manifestations for RPs are the PDF file for a *Paper*, a CSV file containing the raw data observed by a sensor for a *Dataset*, an xml file containing the steps of a certain elaboration for a *Workflow* and so on. Examples of manifestations for RAs instead are their embodiment into a web browser page (see Fig. 4.10) or into a mobile App.

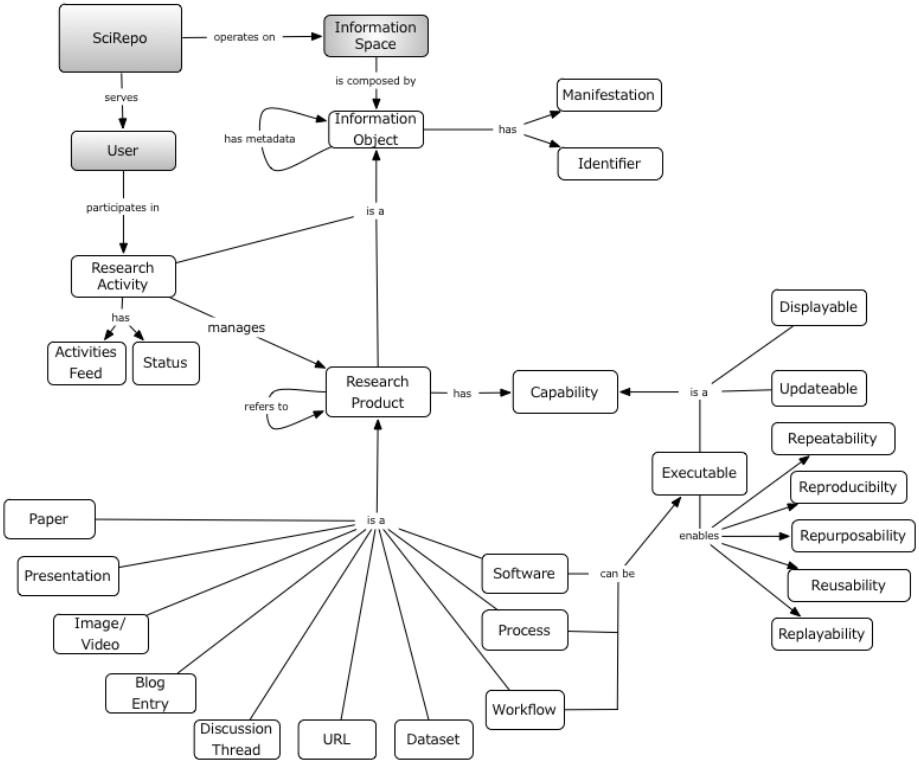


Figure 3.4: The SciRepo End-user Information Space Perspective

Fig. 3.4 further defines the constituents and relationships of an Information Object. According to it, each *Research Activity* has a *Status*, an *Activities Feed* and manages *Research Products*. *Status* represents the current status of the *Research*

Activity, for instance if is completed or ongoing. *User* participates in Research Activities and their interactions and actions are gathered into an *Activities Feed*; an aggregated collection of activities streamed in a chronological reverse order reflecting users' interactions. *Research Activity* manages *Research Product* (RP); any output of the research process is potentially a relevant RP, as such it may be subject of publishing and be related to other products and Research Activities in time and semantics. We classify these outputs into ten separate subclasses ranging from static outcomes like a *Paper*, a *Presentation* or an *Image* to more dynamic ones like a *Dataset*, *Workflow* (experiment) and its instanciation into a *Process*, but also *Blog Entry*, *Discussion Thread*, *URL*, and *Software*. According to Fig. 3.4, each *Research Product* can refer to other Research Products that may, or may not, belong to the same *Research Activity*. Further, a RP has *Capabilities*. The capabilities strongly depend on *(i)* the features supported by the underlying ICT RI Services, and on *(ii)* the subclass/type of a RP, which impacts on them. We identified three kinds of capabilities describing what users can expect from RPs.

A RP could be capable of being *Displayable*: suppose researchers run experiments by executing a workflow collecting input data eventually creating a map, whose high-resolution representation is stored as RP in SciRepo. Then this RP may have the capability to display the map in a RI ICT Service, for example a Geographic information system (GIS) Service which would allow users to view the generated map, and therefore to display the RP, into the GIS service of the RI.

A RP could be capable of being *Updatable*: as we have seen in the previous SciRepo supports both manual and automatic deposition, in latter case RPs have the updatable capability, meaning that RI ICT Services can update it by generating new versions of the same RP.

A RP could be capable of being *Executable*: the capability for a RP to be executable is certainly key for the validation of research result. The executable capability, however, is apt only for three specific types of RPs, namely *Software*, *Process*, and *Workflow* and fits particularly well when we talk about experiments i.e. workflows and processes. It is important to note that SciRepo is not meant to execute these RPs by itself, rather it delegates such feature to the ICT services of the underlying RI providing them with the necessary information they need; this is based on the assumption that if RI ICT services have executed such products before then they may have the possibility to execute them again. As Figure 3.4 shows, we distinguished five subclasses within the executable capability domain, namely *Repeatability*, *Reproducibility*, *Reporpusability*, *Reusability*, and *Replayability* [18]. A RP may have the *Repeatability* capability, meaning that it can be run again after some time has passed, even months or years later, with the same data and configuration. A RP may have the *Reproducibility* capability, meaning that its results can be reproduced with the same data and a different configuration. A RP may have the *Reporpusability* capability, meaning that it can be run again after some time has passed with different data and same configuration. A RP may have the *Reusability* capability, meaning that some of its part can be included into other experiments. Finally, a RP may have the *Replayability* capability, meaning that it not only can be repeated (run again with same data and same configuration) but it is also capable of providing its users with a complete record of it, allowing to go back and forth in the experiments timeline.

RAs are created and staged by SciRepo Administrators, as such we give further details about this process in the SciRepo Administrator perspective part. From an End-user perspective it is however important to note that each RA has its group of users associated. As Fig. 3.5 shows, RA users can be logically divided in three categories: content consumers, content providers and RA managers. In the rest we further describe these roles and the associated functionalities.

Functionality

All the tasks carried out by the SciRepo users are performed by invoking the available functionality. The purpose of the model is not only introduce the potential functionalities but also to identify the ones which are mandatory for any SciRepo. Other specific application domains may consider a number of other non mentioned functionalities as mandatory to satisfy the needs of their user communities. As Fig. 3.5 shows the SciRepo end-user functionalities can be classified in four categories: Personalisation, Access, RPs Management and RA management.

The *Personalisation* class models the functionality that provides the mechanisms for the *Subscription* to existing Information Objects and the request of new RA creation. *Subscription* is about enabling users to be notified about events related to Information Objects. Users can subscribe to events explicitly, i.e., by activating the subscribe function or implicitly i.e. by liking or commenting on an existing RA or RP. Subscribing implies *Notification* of events. End-users can subscribe to different type of events occurring in SciRepo e.g., new product creation, new version products, comments on products etc. Once subscribed they can decide how to be notified about a given RA or RP by accessing the *Notification Settings* i.e. a list of all the RPs and RAs he/she is subscribed where they can select the “channels” through which they should be notified (e.g., SciRepo, email, social platforms e.g., Twitter or none). Finally, any user can exploit the *Request RA* function to request creation of new SciRepo Research Activities. This request is sent to SciRepo Administrators that, if approved, assign the requesting user the role of *RA Manager* for the newly created RA.

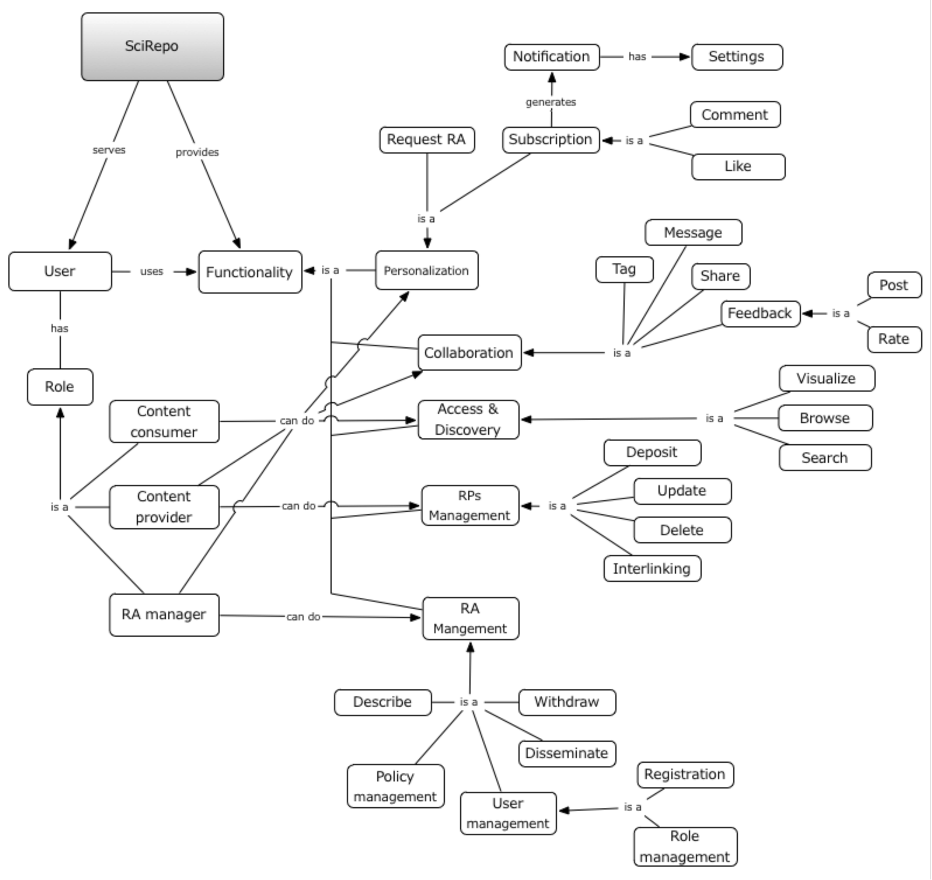


Figure 3.5: The SciRepo End-user Functional Perspective

The *Collaboration* class models the functionality that provides the collaboration oriented part of SciRepo. This functionality relies on social tools (e.g., likes, discussion threads, tags) that record access logs, analytics and accounting (e.g., altmetrics). It comprises the sharing, messaging, tagging and the feedback on RAs and their RPs. This functionality is at the base of the Quality Assessment publishing phase adopted in SciRepo (cf. Sec 3.3.2). *Share* models the functionality that enables user to make RPs noticeable by others outsidethe SciRepo; it allows the user to export RPs via standard protocols and APIs (RDF, Linked Data, OAI-ORE) and towards third-party systems e.g., marketplace repositories, but also scientific social networking (and non) applications e.g., ResearchGate, Academia.edu, Orcid, Twitter. *Messaging* provide means for the user to participate, interact and contact the other users (email-like or instant messaging). *Tag* models the functionality that provides labelling to existing RA or RPs. A Tag is generally a text representing extra information associated to an object, meant to add personal notes about it for future use. In the SciRepo context tags go well further than this, they can be used (by content consumers) to tag other Users or RPs, and this results in involving tagged users or the RP users in the Research Activity with the possibility to give personal contributes. Finally, the *Feedback* function is composed by two important functions. Specifically, the *Rate* function permits end-users to express their position with respect to the ongoing activities and provides alternative form of quality assessment to the RA and/or its RPs. The Research Activity web page in Fig. 4.10 shows a rate example, from 1 to 5, for the two example posts (in the Comments and Discussions part). The *post* function provides users with possibility to add comments to existing RAs or RPs. The Research Activity web page in Fig. 4.10 shows two post examples by two users (in the Comments and Discussions part). When users post in existing RA pages, SciRepo makes these posts available to every user according to his/her preferences. In addition, it enables users to comment, subscribe or re-share these posts. Note that posts are themselves a special typology of RPs of the RA and are indeed searchable and browsable as explained in the previous. It is envisaged that services and applications of the underlying RI can post too, this is the case of Application Posts (APs). APs make users access the subject of the post directly in the application or service used to create it, e.g., to see a posted dataset on the application context which generated it.

The *Access and Discovery* class models the functionalities that provides the mechanisms to consume the RAs and their RPs. It comprises the functionality allowing the discovery of and the access to, on these objects. The *Search* function allows to search existing RAs or RPs, enabling users to discover them, if any, and is capable to fulfil the information need expressed by a user through queries. It represents the access path to the content available into SciRepo and must be customised to fulfil the user requirements. Its services are based on indexes that exploit the RAs or RPs metadata and, in turn, the related provenance information. Once discovered, the product is consumed by means of a *Visualise* function that produces a human understandable visualisation of it. In addition, this function produces visualisation of the profiles describing the users partaking in SciRepo. The *Browse* function provides access to existing RAs or RPs by listing them accordingly to a certain characteristic. It represents a functionality allowing the user to explore the SciRepo content as a whole in the style of a catalog. It may be considered a pre-search mechanism, aiming at finding information useful for searching.

The *RPs Management* class models the functionality providing the mechanisms to populate the SciRepo information space. It comprises the functionality allowing the deposit, the update and the deletion of RPs. The *deposit* function provides the mechanisms to ingest RPs. SciRepo offer both automatic deposit, i.e. in the style of RI services usage, and manual deposit, i.e. in the style of marketplace repositories. In the latter case, examples are not only traditional publications, but also alternative science products, such as web sites, blogs, slides, documentation, manuals, etc. Manual deposit allows scientists to complete the action of publishing a research activity with all the products that are generated out of the boundaries of the RI and connected to it. The Deposit functionality is part of the three publishing phases presented in Section 2.1, as such it deserves an in-depth analysis which is given in 3.3.1. The *update* function allows to modify an already existing RP and is associated with accessibility policies that regulate who and under which condition can update them. This functionality potentially generates the same alteration of the information space as the manual deposit functionality and therefore it implies authoring capabilities. It allows to rearrange the RPs and produce a novel product that may be a newer version of existing one. The *delete* function allows to delete an already existing RP. Even in this case the functionality is associated with policies that regulate who and under which condition a RP can be deleted.

The *RA Management* class models the functionality regulating the “RA life” through the administration of its products and its users. It includes the functionality to *Describe* a Research Activity, by adding a description and data that can be used to enhance its understandings; to *Disseminate* a Research Activity, by exporting it together with its RPs (where applicable) towards marketplace Repositories living outside the boundaries of SciRepo; to *Withdraw* a Research Activity, by explicitly removing it together with its related products from the SciRepo. All these functions can be performed only by users associated with the RA manager role. RA managers also exploit the *User Management* function to perform the *Registration* of new users and their *Role Management*. They are also entitled to exploit Policy Management functions to define the rules governing the RA. Among these functions it is important to note that the RA Manager can set policies for roles and associate these roles either to humans or hooks (cf. Sec 3.2.2).

User

The user dimension represents an important asset of a SciRepo since it clearly identify the actors entitled to interact with it. In fact, SciRepo connects scientists with information during their research activities by supporting the production of new knowledge and the consumption of the already available RPs. Fig.3.3 shows the SciRepo part involving the user concept map. According to the map each User *(i)* is represented by a *User Profile,i.e.,* the descriptive information SciRepo maintains about a single user, *(ii)* uses the *Functionality* as described in 3.5, and *(iii)* is organised in *Group*, *i.e.,* a number of users that are considered or classed together. In addition, each User has *(iv)* an *Identifier, i.e.,* the minimal information enabling to distinguish one user from all the others within an identification scope and *(v)* a *Role, i.e.,* a job function within the context of the RI, with some associated semantics regarding the authority and responsibility conferred on the user assigned role. In the context of the SciRepo end-user perspective, we identified three roles that any SciRepo should support, namely content consumer, content provider and RA manager.

The *content consumer* role is limited to the use of the access functions previously described. As a consequence, can search, browse and visualise RPs within the RA but also Tag and provide feedback on existing RPs by using the Rate and Post functions.

The *content provider* is generally an active participant of the Research Infrastructure. This actor either performs experiments within the Research Infrastructure that yield to the automatic deposition of new RPs or can deposit them manually (cf. Sec 3.3.1). It is envisaged that the update and delete functions belong to this role too.

The *RA manager* is a key role in any SciRepo instance and manages the RA and coordinates the other users by registering them and assigning them roles. Is also in charge of providing a description for the current RA and can disseminate it externally towards traditional marketplace services. This manager has also the possibility to withdraw his RAs. Although we think that withdrawal of RAs should never happen, we reckon that some cases may exist in which one would need to withdraw them.

Policy

Policies are the mechanism used to regulate and restrict the SciRepo access and usage to authorised users. Various approaches exist in designing access control mechanisms, e. g. Discretionary Access Control (DAC) [52], Mandatory Access Control (MAC) [65], Role-Based Access Control (RBAC) [49]. In modeling the access control we used the RBAC approach.

According to the Policy concept map depicted in Fig. 3.3 a *Policy* is a triple (role, functionality, information object). A policy can be indeed associated to a RA of any of its RP and is used to moderate the usage of the single functionality to an established role.

Further, as functionalities act both on RAs and RPs and are used by users the policy must be able to identify the objects that are affected by the functionality in order to provide an effective and fine grained access control mechanism.

3.2.2 The SciRepo Administrator Perspective

The SciRepo Administrator perspective is focused on the *SciRepo Management Functionality* concept. The *User* dimension in this case is represented by SciRepo Administrators only, which exploit this set of functionalities to set up and maintain a SciRepo. The result of this activity is the definition of the most appropriate SciRepo architecture. To perform this task the SciRepo Administrator selects the appropriate components, assigns components to hosting nodes, configures each component including the *Hooking Layer* one and monitors the resulting SciRepo deployment.

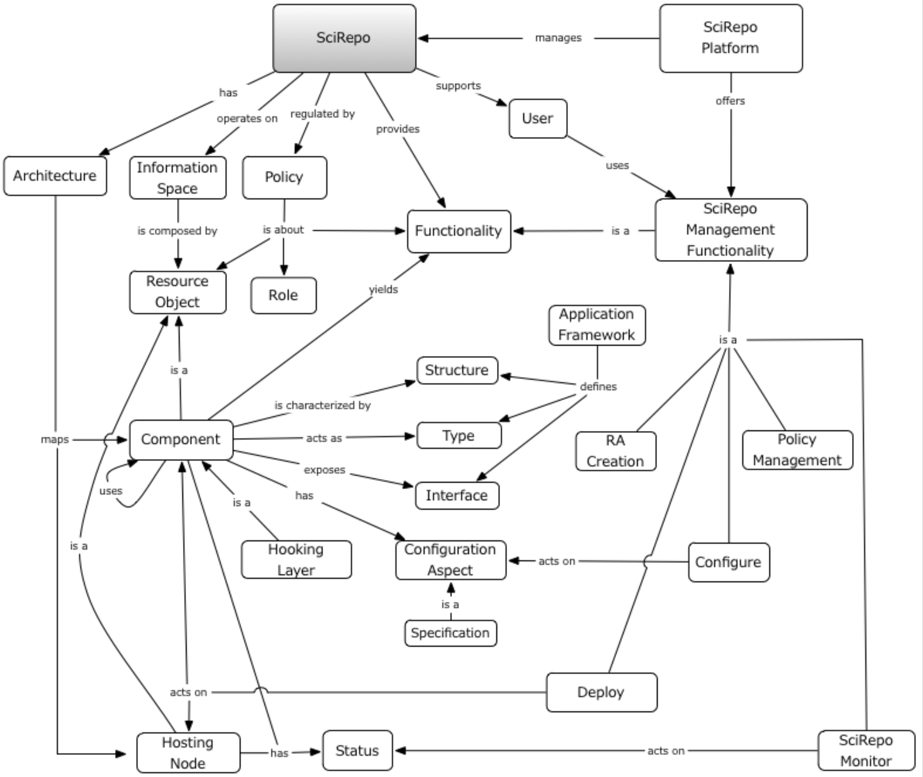


Figure 3.6: The SciRepo Administrator Perspective

Figure 3.6 shows the concepts and relationships of the model that represent the SciRepo Platform from the SciRepo administrator perspective.

Information Space

The Information Space perceived by SciRepo administrators is composed by, and limited to, *Component* and *Hosting Node*. Each *Component* and *Hosting Node* have an unique Identifier and associated Metadata.

Specifically, *Hosting Node* identifies the hardware devices providing computational and storage capabilities. It is characterised by the following properties: *(i)* it is connected to the network; *(ii)* it is available in the infrastructure; and *(iii)* it is capable to host software components. Its characteristics are expressed through a *Profile* tailored to report the hardware architecture, the operating system, the environment, the available storage, and the installed software packages.

A *Component* is the software package that may assume the status of web service, web resource, or module to deliver a set of related functionalities. It is autonomously configurable and deployable on one or more hosting nodes. It exposes its capabilities through interfaces that represent the signature of the component, while its logic is entirely encapsulated. They are therefore substitutable since a component is equivalent to another if the interfaces and the encapsulated behaviour are the same even if their implementation are completely different. The component characteristics are expressed through a *Description* tailored to simplify a correct and appropriate use of the component. This description may range from a human oriented description, e.g., a textual description in natural language, to a machine understandable one, e.g., the WSDL of a web service. A component can interact with other components to deliver its functionality, either hosted on the same hosting node or distributed over other hosting nodes belonging to the same network. In addition, when deployed a component may have a *Status* that expresses set of values of all the parameters that define its condition.

Functionality

From the SciRepo Administrator perspective the *SciRepo Management Functionality* provided by the SciRepo Platform is entirely related with the set up of the architecture for a SciRepo by means of the configuration, deployment, and management of its constituent parts.

Specifically, the *Configure* function allows a SciRepo Administrator to act on the configuration aspects of a component in order to modify/customize its behaviour and thus the provided functionality. This task prepares a component *(i)* to be activated on a specific hosting node; and *(ii)* to be aware of the characteristics of the RPs it has to manage.

The *Deploy* function enables the SciRepo Administrator to enact a function by assigning a component to a hosting node and make it capable to operate, i.e. to provide the functionality the component is implemented for.

The *RA Creation* function enables the SciRepo Administrator to stage new Research Activities in SciRepo. The creation of new RAs can be also based on requests performed by end-users. In this case Administrators create the RA and successively assign the RA Manager role to the requesting user throughout the *Policy Management* function.

Finally, the *SciRepo Monitor* function allows the SciRepo Administrator to monitor the deployed component status. Usually this functionality allows supervising the average number of requests managed by the component, the average load of the hosting node, the average number of queued requests, the latency, the throughput, etc.

Architecture

As introduced in Section 3.2, the architecture is a representation of the system dealing with mapping functionality onto hardware and software components. Our model is based on the understanding that *Components* and *Hosting Nodes* are the building blocks of the SciRepo Platform and, that, in order to allow them to operate as an application, an *Application Framework* is needed.

The Application Framework models the environment each component is conceived to work in. It defines the component *Structure* and its *Type* and it identifies the *Interface* to which components have to conform to interact, thus prescribing the component to component interaction patterns. Because of this, an application framework plays a fundamental role to enable systems as federation of enacted components that exchange meaningful and context-driven data. Each component, when considered in isolation has its own application framework. To build a distributed system among two or more components, the application frameworks surrounding them should either be interoperable or be reconciled to some extent. The heterogeneity to be reconciled may vary from the standard or protocol exploited to implement a certain functionality (e.g., the component-to-component communication standard) to the need for having a component implementing a certain functionality in a specific way (e.g., a component exposing its facilities through a certain interface). The two reconciliation approaches have both to be considered and each of them identifies a number of properties that have to be described to let the SciRepo Management Functionality to work properly.

The *Structure* concept models: *Lifetime*, intended to define the deployment, activation, update, and failure management; the *State*, intended to model the context, business, and session data persistence; the *Configuration*, intended to define the properties that can be specified to alter the component behaviour.

The *Type* concept models the communication approach the component is designed to work with. It includes: *Service*, intended to model a component encapsulating a number of functionalities; *Blackboard*, intended to support asynchronous communication between components where one component write information in the blackboard and another one (or other ones) read those information from it; *Adaptor*, intended to model a component that acts on behalf of another one to translate and enrich its interface in/to a different one; *Proxy*, intended to model a component that passes unmodified requests to another one; *Mediator*, intended to provide a unified interface to a set of other component interfaces while potentially enriching the exchanged information; *Broker*, intended to select the most suitable component serving the requested functionality.

The *Interface* concept models: *Messaging-oriented specifications*, intended to give a framework for exchanging information in a distributed system, e.g., Simple Object Access Protocol (SOAP), Web Service Addressing (WS-Addressing), Web Service Notification (WS-Notification); *Description specifications*, intended to define the information sufficient to permit the exploitation of the component, e.g., Universal Description Discovery & Integration (UDDI), Web Services Description Language (WSDL), Web Services Resource Framework (WSRF); *Transaction and Security specifications*, intended to provide reliable messaging, coordinated behaviour, and secure communications, e.g., Web Services Security, Web Services Secure Conversation Language, Security Assertion Markup Language, Web Services Reliable Messaging; *Application specifications*, intended to define specific and contextualised behaviour for the exchange of information, e.g., Open Archives Initiative - Protocol for Metadata Harvesting (OAI-PMH), Open Archives Initiative Protocol - Object Exchange and Reuse (OAI-ORE), Search/Retrieval via URL (SRU), Open Geospatial Consortium Web Map/Feature/Coverage/Processing services (WMS, WFS, WCS, WPS).

3.2.3 The RI Application Developer Perspective

The RI Application Developer perspective concerns the actors in charge of extending the RI ICT services and applications implementing the hooks to connect a RI. This activity is performed by exploiting the APIs. As Fig. 3.1 shows, the RI Application Developers does not see neither the SciRepo internals nor the SciRepo functionalities as perceived by the end-users, rather they perceive it throughout its lower layered component: the *Hooking Layer* component. This component is the enabling core component of the SciRepo Platform, it is the bridge connecting any RI service to

SciRepo capabilities and it is in charge of populating the SciRepo content automatically during the research life-cycle (without scientists being directly involved in the actual action of publishing).

The RI developer interacts with the Hooking Layer by implementing specific RI programs/scripts, namely hooks. An *Hook* reacts to RI events (e.g., dataset creation) by calling the Hooking Layer API to transform these events in meaningful information (e.g., dataset deposition in SciRepo).

Figure 3.7 shows the concepts and relationships of the model that represent the SciRepo from the RI Application Developer perspective. According to the model a SciRepo interfaces with the RI through the Hooking Layer. This layer provides *functionality* acting on the *Information Space,* which is modeled by a graph (similarly to a classic entity relationship diagram) and all of this is regulated by policies.

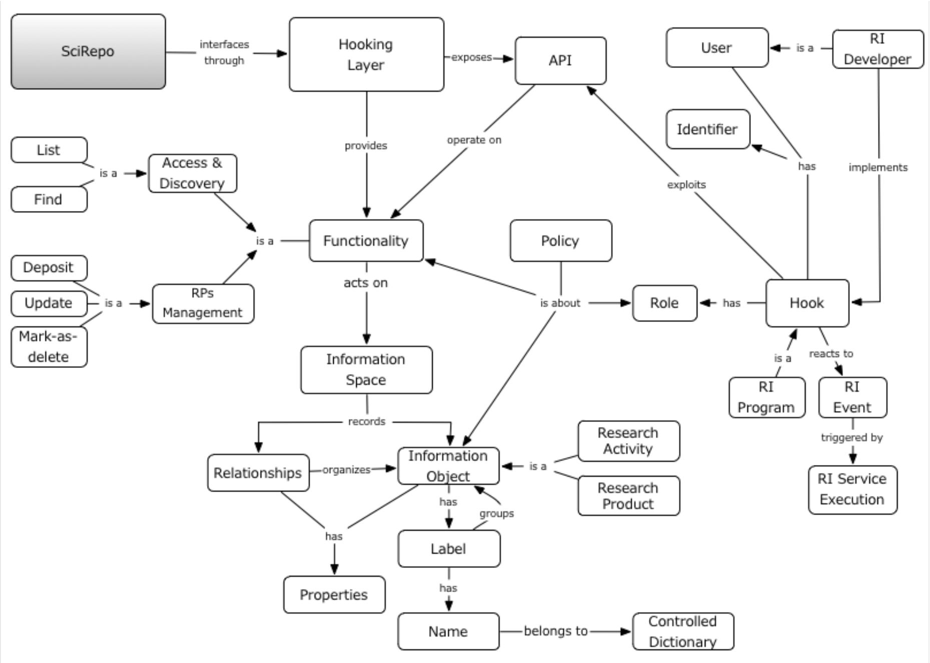


Figure 3.7: The RI Application Developer perspective

In the following we present the concepts and relationships of the model starting from the and Information Space, Functionality, User and Policy main concepts.

Information Space

The *Information Space* within this perspective is modeled as a *Graph* similarly to a classic entity relationship diagram. The property graph contains connected entities (the nodes) which we mapped onto four different categories: Research Activity, Research Product, Scientific Result (e.g., the end of a process concluding successfully or not) and RI Resource (e.g., a resource of the Research Infrastructure exploited in some research activity that need to be referred). The Hooking Layer API eventually acts on the SciRepo *Information Space.*, for this reason the RI Application Developer needs to have a proper knowledge of the concepts behind this graph, which is certainly one of SciRepo main features.

The *Information Object* concept can hold any number of metadata in the form of attributes (key-value-pairs). Information Objects can be tagged with labels, for contextualizing them in the RA domain. In fact, a *Label* is an optional addition to the graph that allows to group Information Objects into sets. All Information Objects labeled with the same label belongs to the same set. An Information Object node may be labeled with one or more labels.

The Relationship concept models directed and, semantically qualified connections between two or more Information Objects. A relationship always has a direction, a type, a start Information Object, and an end Information Object. Just the same as Information Objects, relationships can have any number of *properties*. It is important to notice that, even though relationships are directed, the reverse relationship is always hold implicitly (for the sake of readability and clearness they are not reported) thus allowing the reverse connection between the end Information Object and the start Information Object. This is particularly important, for example, to know which is the process that generated a given dataset. Just like in entity relationship model, the SciRepo must ensure the “No broken links” property, meaning that RI Application Developers cannot delete an Information Object without also deleting its associated relationships. On the other hand, they can also always presume that an existing relationship will never point to a not existing Information Object.

Functionality

The RI Application Developer invokes the Hooking Layer functionalities by exploiting the API exposed by them. As Fig 3.7 shows the RI Application Developer functionality can be classified in two categories: Access and RPs Management.

The *Access* class models the functionality that provides the mechanisms for the identification of RAs and their RPs. It comprises the functionality allowing the “programmatic” discovery of and the access to of these objects. The *Find* function allows to look for existing Information Objects and obtain relevant information about them, *i.e.,* RA/RP object instances containing properties characterizing them. Find allows developers to implement "stateless" hooks. For example, an hook needing to publish a RP could choose whether to create a new RP or update it version by exploiting such function. The *List* function provides access to a listing of existing Information Objects the hook is entitled to operate with. It is important to note the these functionalities are associated with accessibility policies that regulate what and under which condition can update them.

The *RPs Management* class models the functionality that provides the mechanisms to populate the SciRepo content. It comprises the functionality allowing the deposit, the update and the deletion of RPs. The *deposit* function provides the mechanisms to ingest new RPs. This is how SciRepo implements the automatic deposition of RPs (cf. Sec. 3.3.1). The deposition of an Information Object may be implemented either as a copy of the object generated in the RI or a registration of the object that links the SciRepo virtual object to the physical object at access time. In the former case, it is up to the developer to ensure to enrich the object with the proper metadata, e.g. provenance. In the latter case, it is up to the developer to enrich the object with the proper information to access it at runtime, e.g. type of connection and supported security token. The *update* function allows to modify an already existing RP and, even in this case, is associated with accessibility policies that regulate who and under which condition can update them. The *mark-as-delete* function allows to mark an existing RP for deletion. This is needed to handle cases where, for some reason, a deposit operation went wrong. Once marked a RP for deletion, a notification is sent to the SciRepo RA Administrator who reviews the delete request and, if necessary, actually deletes the RP from SciRepo.

User

The user dimension within the Hooking Layer is represented by the actors entitled to implement programs interacting with it. Fig 3.7 shows the part involving the user concept map. Each User *(i)* is a RI developer who implements hooks (these exploits the API to enact the exposed functionalities), *(ii)* must be an identified user of the SciRepo and must possess the required credentials to access it, *(iii)* has a *Role* that defines on which functionality and research activity he/she can operates on.

Policy

Policies are the mechanism used to regulate and restrict the Hooking Layer, and consequently the SciRepo, access and usage to authorized developers. Even in this case, for modeling the access control we used the RBAC approach [49].

According to the Policy concept map depicted in Fig. 3.7 a *Policy* is a triple (role, functionality, information object). A policy restricts the use of a single functionality to an established role for a given RA. A role defines which hooks are supported in a RA, thus implicitly identifying which RIs can be connected to a RA. Moreover, by simply defying different policies, it can be ensured that in a RA the RI objects are persistent since it is denied to implement updates, while in another it is possible to modify existing object with newer versions. It is important to notice that within this perspective Roles are associated to hooks and not to users (since they are the entities using the functionalities). Therefore, each hook has an identifier in a SciRepo, and is associated to a Policy which defines what functionality it can use and on what Information Object.

3.3 Publishing Cycle in SciRepos

We have seen that the combination of integration with RI, rights and quality information about products introduces a novel publishing paradigm where "publishing" is intended as making a product online available, discoverable, peer-reviewable, reusable according to given rights, real-time accessible, citable, and interlinked with its research activity and associated products. Possibly according to the FAIR[1] principles.

In this section, we shall present the publishing phases (Deposition, Quality Assessment, Dissemination) as realised by a SciRepo, accompanied by the relative benefits in terms of overcoming the aforementioned methodological drawbacks (cf. Sec. 2.2.1).

3.3.1 Deposition

SciRepo offer both automatic deposition, i.e. in the style of RI services usage, and manual deposition, i.e. in the style of marketplace repositories.

Automated publishing is achieved by connecting the SciRepo with the underlying ICT services of the RI, in order to intercept the creation of products, publish them, and notify interested scientists of this. Such integration is part of the design of the SciRepo, is based on the publishing needs of the given RI community, and manifests in the implementation of the hooking layer envisaged. For example, the community may be willing to publish as a product new datasets generated by experiment into ICT storage services, or publish the execution of an experiment and the relative results (i.e. application of an algorithms over given input data, together with resulting data). To make this possible ICT services should communicate with the SciRepo to notify that products have been produced within a research activity context, with a given unique web resolvable identifier, and metadata description. Notification of products creation translates in SciRepo events, highlighted to scientists via social tools as posts of type "publishing". Scientists find in the post the link required to access the original data, can start a thread of discussion about the post, can forward the post according to standard practices.

Publishing can also occur manually, typically for all products relative to a research activity that are not automatically produced by the RI ICT services during experimentation. For example, interesting web sites created by scientists, threads of discussions in online blogs, technical documentations, software, scientific publications, datasets produced out of the RI boundaries, etc. In all such cases, scientists access the SciRepo and deposit under a given research activity a product of a given type,

[1] Force11 - FAIR principles: https://www.force11.org/group/fairgroup/fairprinciples

together with descriptive metadata. The product can be deposited locally or just be referred to from where it is online accessible. As such, this action resembles deposition operations typical of publications and datasets repositories, with the important difference that products are ingested in the context of a given research activity and are notified to scientists with a post. As such, they are implicitly linked to all products of the research activity and also associated to one or more discussion threads in the SciRepo.

3.3.2 Quality Assessment

SciRepos should support both traditional forms of single/double blind peer-review and alternative forms of peer-review, counting on social tools (e.g., likes, discussion threads, marks) and underlying RI ICT services that can (i) automatically verify and rank quality or compliance of products to agreed community quality indicators (e.g., dataset conforming to standard formats, within given size or value ranges), or (ii) record access logs, analytics and accounting (e.g., altmetrics).

3.3.3 Dissemination

When deposited, products are assigned a unique web resolvable identifier and may be associated with different kinds of metadata descriptions. Such descriptions may be gathered by underlying ICT services (e.g., provenance, authorship) or be specified by scientists, and discerns products by their typology (to be decided by RI scientists).

Their nature depends on the RI at hand and typically enables discovery and reuse of products (e.g., interpretation, citation, access rights) at different level of scope (e.g., experiment, research activity, RI, outside the RI). Similarly, relationships between products can be collected by ICT services while performing experiments and creating products (e.g. versionOf, relatedWith, likedByScientist, discussedByScientist) or be specified by scientists via SciRepo user interface. As specified above, access to products should be ruled by proper right management tools, which may authorise scientists to access products based on their role in the RI (e.g. groups of scientists), the research activity at hand, the typology of products, and the quality of products (e.g., products of low quality are not “published” to given groups of scientists). Finally, once scientists discover the products they are interested in, the SciRepo enables the set of actions they are authorised to fire, based on product typology and user rights. The complexity of such actions depends on the SciRepo implementation and its embedding within ICT services. For example scientists may be authorised to visualise or download a product (e.g., a publication PDF), re-execute a product (e.g., an experiment), link a product to another, interact with the history of versions of a product, open a discussion about a product, or review a product.

3.3.4 Publishing in SciRepos: the Benefits

In the following we describe the benefits resulting from SciRepos. We focus on how the SciRepo deposition, quality assessment and dissemination phases overcome the drawbacks of current publishing practices introduced in Sec. 2.2.1.

Benefits during Deposition phase

- **In context:** products are fully fledged, i.e. they are linked to the entire setting leading to them. Products that are published in marketplace sources, such as datasets, software, or scientific publications, can be manually re-connected to their research activities hence be discovered in-context together with links to related products;

- **Products remain "alive":** products may change after they have been published, scientists are actually using a reference to have access to them. Besides, they are expected to be dynamically versioned so that it is always possible to have access to the instance of a resource at a certain point in time;

- **No extra cost:** products are stored in the underlying ICT services, where they are created and managed, hence costs and risks of moving products outside RI boundaries are dropped.

- **Alternative products:** support to alternative products, which can find in SciRepos a place where they can be manually deposited, evaluated (e.g. social "like" tools, discussion threads), discovered, and linked to a research activity or other products.

Benefits during Quality Assessment phase

- **Continuous and in context:** published products can be continuously assessed from a qualitative perspective, e.g., it is always possible to annotate any research product with a comment (also a process) aiming at demonstrating either the outstanding nature or the mediocrity of the product. The scientific context where the product is created and possibly reused is the best qualified to assess the quality of the product since it represents the primary target domain to be served;

- **Self-assessment:** the RI community has the ability and interest to define "certificates of quality" which are crucial to enable scientific reward mechanisms pertaining non-traditional products. For example, a web blog kept by a scientist and considered a reference for other RI scientists may be published in the SciRepo, be peer-reviewed and certificated as high-quality research, and such awards be spent to enrich the scientist's CV.

Benefits during Dissemination phase

- **Unified:** Scientists can be offered marketplace facilities to discover and access products that subsume those typically offered by publication and dataset repositories; they can discover objects by typology, cross-typology, navigate their relationships, configure their access rights, and profit from advanced online re-use functionalities;

- **Automatic and complete:** product authors are less burdened by tedious activities of metadata information and relationship curation.

4

A SciRepo Platform Reference Architecture

A Reference Architecture is a template solution that maps the functionalities defined in the Reference Model (cf. Sec. 3.2) onto software components implementing them. It is not based on a specific technology, rather it provides guidance for architecture principles and best practices providing an architecture baseline and an architecture blueprint.

The SciRepo Platform Reference Architecture that we have designed and present in this chapter, is an architectural philosophy that promotes the development of SciRepo applications not as single units with tightly coupled business logic rather as loosely coupled components, each of which performs a logical, discrete function. A concrete architecture for a SciRepo Platform can be derived by following it together with its architectural patterns which incorporates several application scenario components.

4.1 Requirements

In 3.2 we presented the reference model explaining the functions a SciRepo platform has to provide to its users. For each function we provided a detailed description and described how it relates with the other platform concepts. To these functions we now add number of important quality attributes for a SciRepo platform, which are given in the following and in the form of *non functional requirements:*

- **Interoperability:** any SciRepo architecture must be designed to be interoperable with the underlying RI ICT services and with the external marketplace repositories or third-party applications. SciRepos has to be able to store or refer different traditional types of content (i.e. papers) as well as complex other non-traditional entities (i.e. datasets, workflows, blogs) . While each form of content has unique aspects, it is desirable to manage this content in a uniform way. For this reason SciRepo should adopt well known metadata formats in order to improve interoperability.

- ***Reliability:*** Reliability, in the meaning of ability of a system to keep operating over time is a peculiar aspect of any SciRepo Platform. There two aspects to take into account for this requirement: the RI reliability and the SciRepo reliability. Assuming that different organisations partaking in the Research Infrastructure may have different reliability requirements for their set of services, RI reliability is about making sure that the underlying set of services are obtained from a reliable provider and that a level of trust in the service's accuracy and reliability can be established. SciRepo reliability means that it operates correctly and either does not fail or details any failure to the administrators or users. An aspect to consider for SciRepo reliability is *atomicity*. If one upgrade its Information Space, it is desirable to make this operation as atomic as possible; the SciRepo platform should be either in the new state or in the old state, but certainly not somewhere in the middle.

- **Security**: Security may indicate different things with respect to software systems, in a SciRepo context should be associated to the following characteristics:

  – *Authentication*: should support from a simple mechanism based on username and password to secure delegated access provided by third-parties via open standard protocols (e.g., OpenID, OAuth2);

  – *Authorization*: the access to information or service is granted only to authorised subjects. These subjects include uses and services, in the latter case they may have access restrictions based on the identity of the service user, where possible;

  - *Data Protection:* includes the Encryption and Integrity. The first technique is to preserve privacy and normally based on the obfuscation of the content by means of a data stream transformation based on a public-private key mechanism. The integrity make sure that information is not corrupted;

  - *Message Protection*: can rely on the encryption of the transport channel or can require advanced digital encryption, or partial encryption, and intrusion detection and protection.

- **Scalability:** The SciRepo architecture must work without degradation of other quality attributes when it is changed in size or in volume in order to meet users needs. There are two alternatives for solving scalability issues:

  - Scale-up: also known as vertical scalability, means upgrading to more powerful hardware for the service site.

  - Scale-out: also known as horizontal scalability, is about distributing the workload across more nodes. In other words you scale by adding more machines into your pool of resources;

  If addressing scalability presents possible performance issues the source of them has to be identified. Performance is perceived through both the Latency, which measures the delay interval spent between the request of a functionality and the time when the functionality effect begins to perform its task, and the Throughput, which measures the rate at which a functionality sends and receives data, i.e. the number of requests it is capable to serve in a certain interval of time. The SciRepo performance ought to be studied and performance models must be constructed. These models will need to be calibrated based on the option or combination of options is chosen to make sure that the new SciRepo configuration meets the planned scalability requirements without influencing its performance.

- **Consistency:** Consistency, in the meaning of the database systems domain, is one of our most important quality attribute. Since SciRepo functionalities identified in 3.1 divide them in two main categories: Repository-oriented and Collaboration-oriented functionalities, they may have different requirements in terms of consistency model. It is envisaged that Repository-oriented functionalities adopt *strict* *consistency* while the Collaboration oriented ones may opt for *eventual consistency* .

- **Extensibility:** Extensibility for a SciRepo architecture is very important because its “business environment” is continually changing and evolving by definition. The SciRepo architecture must be flexible and adaptable to the changes and seriously take future growth into consideration. One of the major obstacles in this part is to try to adapt existing services without changing the interfaces. Changings in the service interfaces could have a major impact on the success of a SciRepo platform architecture in the medium-long term. Thus, the additional functionalities provided by the developers implementing a concrete architecture for a SciRepo must be easily integrated.

- **Adaptability**: Adaptability is the ease with which a system may be changed to fit changed requirements. The use of a Service Oriented Architecture approach in this case would have various benefits to the ability to adapt. To attain adaptability, the services need to be managed and monitored properly as a single coherent solution, and the communication between the service and the underlying infrastructure must be managed. An actual quantification of performance (which we suggested also for scalability), capacity, and availability is required to implement this management function.

- **Usability**: Usability assesses how easy user interfaces are to use. The word “usability” however also refers to methods for improving ease-of-use during the design process. SciRepo must provide interfaces that support the normal usability operations such as cancelling a request, undoing the last operation, and obtaining information for feedback such as percentage completed and time to completion of functionality operations where necessary.

- **Efficiency:** The last quality attribute of importance is efficiency. Software systems tend to become larger and more complicated. Consequently, it is advisable to make functionalities as more efficient as possible; only what is required to do, should be done. Nevertheless, efficiency may not break the previous quality attributes, as they are more important.

4.2 Architectural View

Software architectures are generally achieved by designing software components into three tiers (three-tier architecture): the Presentation tier, comprising the part of components dealing with interface issues; the functional/process tier, comprising the software implementing the logic proper of the specific application; and the Data tier, which enable final storage of information and further data processing supporting the access to and the persistence for components requiring it.

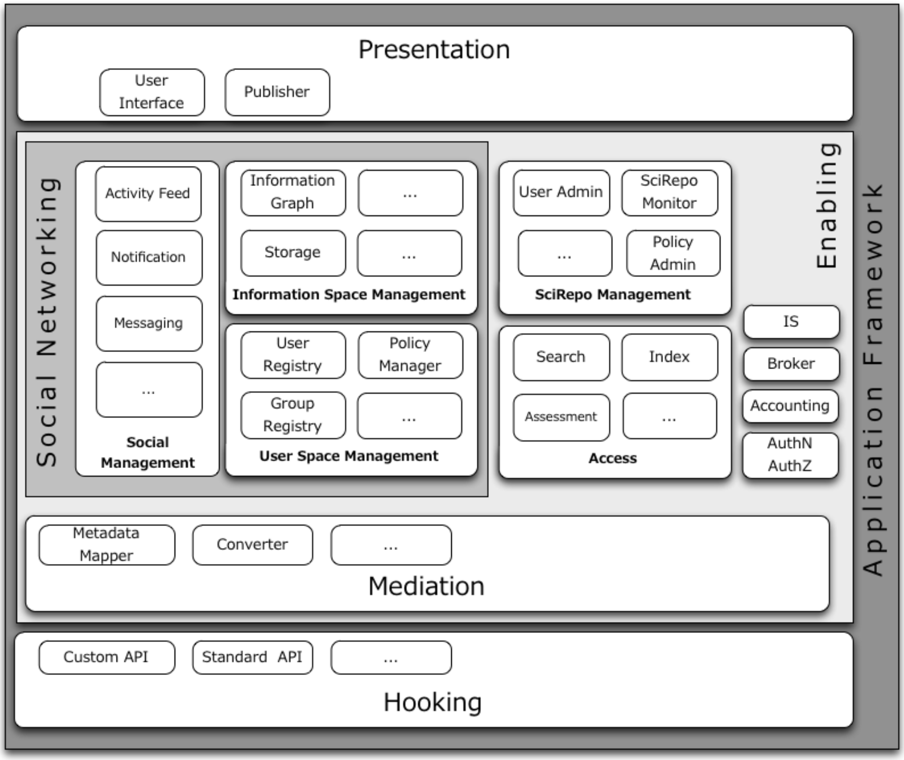


Figure 4.1: A reference architecture for a SciRepo Platform

It is quite common to develop complex applications by splitting components along these tiers, in order to achieve modularity; to realise thin clients, comprising only the presentation layer of a component; or to have the data common to several components centralised in a separate component, which therefore realises the data layer of these components. However, modern software architectures go ahead a mono dimensional layered vision even if the main achievements are maintained and exploited. They are based on use relationships that may exist a priori among any subset of the components; the components can be combined in different ways to support different functionalities; and the same component may be used in different ways, in accordance with the restrictions placed on their use and the goal of use.

Consequently for the SciRepo Platform reference architecture we propose a more detailed software architecture classification as shown in its architectural view provided in Figure 4.1.

Starting from the bottom, any SciRepo Platform is meant to interface with a selection of underlying RI ICT Services, as such it must be equipped with a set of connectors bridging actions operated on resources in the RI with events recognised and exploited by the SciRepo components; we collect this connectors in a functional area called *Hooking*.

Over the Hooking layer, the SciRepo Platform must implement appropriate conversion mechanisms of RI resources created and maintained by RI ICT services (e.g., datasets, processes, workflows, blogs, etc.). These resources are heterogeneous and pre-existing to the SciRepo instance implementation; their structure and format rarely satisfies the rules imposed by components implementing SciRepo end-users functionality; to deal with the format mismatches and heterogeneity we add a functional area called *Mediation*.

Successively, as shown by the architectural view, the core business logic layer of a SciRepo Platform is partitioned into 5 functional areas:

- the *Information Space Management* area that includes the necessary tools providing the management and the storage of SciRepo information and resource objects;
- the *Access* area that deals with the discovery and rating of information products;
- the *SciRepo Management* area that provides all the necessary tools to manage the SciRepo, and to activate all the necessary processes required for the management of the information space;
- the *User Space Management* area that delivers all the necessary mechanisms to manage the user access rights and policies;
- the *Social Management* area to manage the SciRepo social networking facilities.

Finally the Presentation area collects all the components exposing SciRepo functionalities to end-users. This area is not limited to the User Interfaces, it also includes as many as public standard APIs to disseminate the SciRepo content to external marketplace services.

In the rest of this chapter we shall describe the main characteristics of the presented functional areas.

4.3 Components

4.3.1 The Hooking Area

The Hooking area is at the bottom of the layers' stack (see Fig. 4.1), it is the access point to the SciRepo for RI services; consequently it should be furnished with as many as public standard, and de-facto standard, interfaces as possible. According to Figure 4.2 we split the Hooking functional area in two main functions: the *Connector* and the *Connection Management.* The first one wraps all the components necessary to interface with RI service hooks. Among these we envisage standard interface protocols for repositories such as *OAI-ORE* but also a custom set of APIs exposed via an *HTTP REST* interface. The second, as the name suggests, deals with connection management facilities and is composed by three sub-components: *(i)* a *Connection*

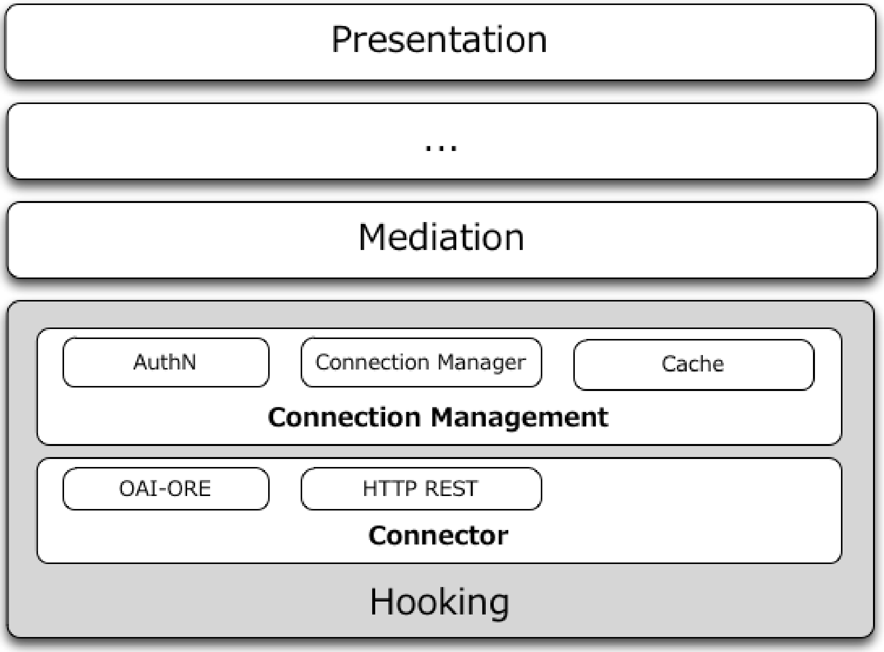


Figure 4.2: The SciRepo Hooking area

Manager component delegated to manage client connections and concurrency; it should use threads for handling client connection requests provided by the connectors and associate each of this request with a thread dedicated to it*, (ii)* a *Cache* component which should be interrogated first to see whether it contains a thread that can be reused for the connection, *i.e.* the manager should create new threads only when necessary, and *(iii)* an *Authentication* component coping with authentication and request processing from the RI service hooks.

4.3.2 The Mediation Area

SciRepos must deal with products that may vary in their structure, format, media, and physical representation. In order to fulfil these requirements a SciRepo Platform must be able to convert these products (coming from the underlying RI services) and make them accessible in the SciRepo. This work is performed by the Mediation Area, that includes a set of components providing the platform with identification, conversion and metadata mapping of heterogeneous and multi-institutional external resources. A graphical representation of these components is presented in Figure 4.3.

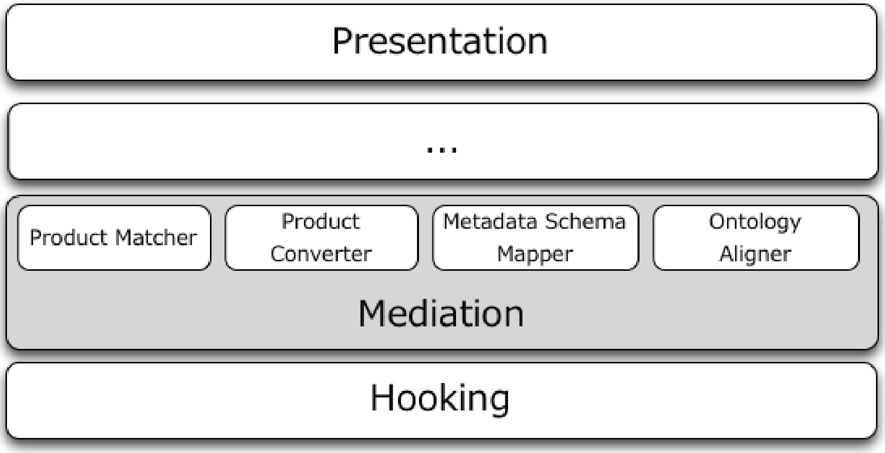


Figure 4.3: The SciRepo Mediation area

The *Product Matcher*, by exploiting encoded knowledge about the API methods used by RI hooks creates a representation of the product matching the information space graph model expected by the other SciRepo components, in other words transform the RI resource in a SciRepo Research Product. The *Product Converter* takes this representation and build the necessary semantic relations to connect the product with the existing Research Products already present in a Research Activity graph node (cf. Fig.3.7). Finally, since dealing with heterogeneous information sources implies also managing multiple metadata formats, the *Metadata Schema Mapper* component is in charge of generating alternative metadata representations according to given metadata schemas, and through exploiting the features of the *Ontology Aligner* component, which identifies and suggests semantic correspondences between the representational elements of heterogeneous ontologies.

4.3.3 The Information Space Management Area

Papers, datasets, workflows, anything that can be mapped into our notion of Information Object (cf. Sec. 3.2) relies on the Information Space component of a SciRepo for its management and persistence. The Information Space Management Area is therefore essential for the correct functioning of the SciRepo Platform. According

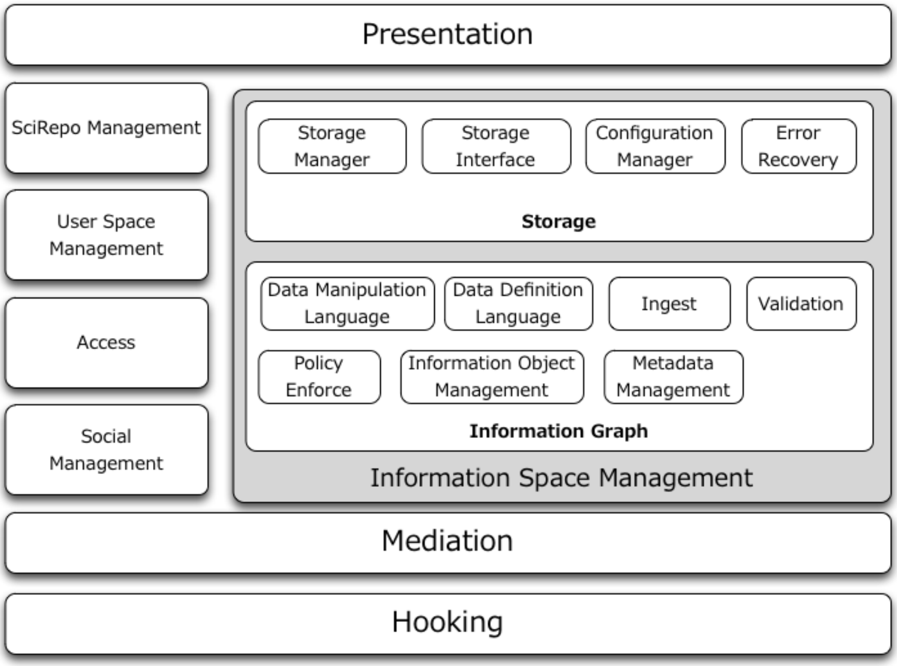


Figure 4.4: The SciRepo Information Space Management area

to Figure 4.4 the set of functions identified for this area is composed by two main components*: (i)* the *Information Graph* component, needed to maintain semantic relationships among the four categories of information objects identified in 3.2.3, which also offers data definition, manipulation, and access languages, and *(ii)* the *Storage* component, providing the mechanisms and facilities for storing the payloads of the information graph when needed.

In the following we describe them and add more details about their functioning.

Information Graph

The *Information Graph* component provides the instruments and functions for the management of SciRepo information objects; these functions are offered in a typical repository fashion, they include management of the information object relationships with other information objects, their versions, the administration and enrichment of the metadata, and mechanisms for their discovery and access. In order to perform this wide set of functions, SciRepo exploits the capabilities of two principal components coordinating and exploiting all the others: the *Ingest* component, allowing to properly add research products and associate them to a research activity, thus including them in the information graph, and the *Information Object Management* component providing capabilities to prepare the products for storage and management.

The Ingest functions include *(i)* receiving the product from the mediation layer, *(ii)* performing quality assurance through the *Validation* component, and (iii) verifying the authorisations of the hook activating the process through the *Policy Enforce* component.

The Information Object Management functions include *(i)* the assignment and/or validation of an unique identifier; this identifier must be unique within the SciRepo, *(ii)* the generation of additional metadata by means of the *Metadata Generator* component, and *(iii)* languages for definition and manipulation of information objects by means of *Data Definition Language* and *Data Manipulation Language* components, respectively.

Storage

The *Storage* component provides the mechanisms and functions for the storage, maintenance and retrieval of information objects. According to Figure 4.4 this component can be logically divided in four functional area. The *Storage Manager*, in charge of transforming the information object from how it was submitted, along with its associated metadata, into a bytestream that can be stored on suitable hardware; this transformation is driven by the *Configuration Manager* component, and can exploit the capabilities provided by the Product Converter component of the Mediation area(cf. Sec. 4.3.2). The detection of the corruptions during transformations or internal data transfer is handled by the *Error Recovery* component that provides statistically acceptable assurance that the bytestreams are valid. Considering that the clients should be able to access the storage content independently of the actual system wrapped by this component, the need of a *Storage Interface*, capable of plugging different back-ends for data storage is essential; a type of back-end may be preferable to another. For example, a column family store could be more suitable for the social networking facilities, a triple store could be more suitable to support the export of RPs via Linked Data, and so on.

4.3.4 The Access Area

The access area contains the components in charge of providing the access functionality depicted in Figure 3.5 and explained in the related section. In particular, in Figure 4.5 we have identified two main components: the *Search* and the *Index* components.

Search

The Search component allows SciRepo users to locate information objects in a cost-efficient manner, while satisfying the level of quality required to be met by the overall data retrieval and delivery operation.

Search operations are initiated by the submission of user queries to the search engine. A number of other components are used to complete these operations. Nevertheless internal functionality handles the overall process, the enforcement of the constraints, the verification of the access rights, and the correct delivery of the results. Moreover, the semantics of the query language allows the simultaneous use of both boolean and probabilistic data processing operators that require the management of query normalisation mechanisms.

A *Query Parser* carries out the typical process of parsing the input (query) while looking up for relevant information on resources such as collections and query language adapters. Dealing with heterogeneous information sources, in fact, imposes the management of multiple metadata formats, content types, and media that requires different Index components. The *Query Adapter* transforms the initial query in a number of queries suitable to be processed by the different Index components.

The *Query Optimizer* attempts to process the output of the parser in order to optimise it. This query-optimisation procedure is accomplished with regards to various performance/cost metrics and makes use of advanced computationally intensive algorithms. The *Query Optimiser* consolidates information provided by various components such as:

- the *RA Selection*, that exploiting the research activity descriptions managed by the *RA Description* component and a given content-based query, reports which RAs are the most appropriate to query and are deemed to contain topically relevant information objects. This selection is based on a multi-criteria decision model, where different parameters in addition to content relevance, such as cost and connection time, are also considered;
- the *Index* component that provides information about the indices, the indexed metadata, the metadata schemas, and the indexed content media types;
- the *Personalization* component that provides information about users and groups, including skills, expertise, preferences on content, services, and quality of service extracted by the user registry.

The result of the *Query Optimiser* is the query execution plan used by the *Query Execution* component to contact the *Index* components in order to complete the query and prepare the result-set. This result provides the actual information that the end-user is presented with: information object descriptors along with partial metadata are ultimately being returned to the end-user process, thus allowing for retrieval of actual information objects or their URIs where possible.

Index

The Index component is the key to speed up the retrieval process. In order to distribute the load of user queries and to parallelise expensive retrieval process, indexes can also be stored at different hosting nodes (replication, partitioning).

Essentially, an index replicates information from primary content resources and organises data so that it is possible to evaluate queries efficiently. For instance, in a database kernel, implicit triggers guarantee the consistency between the tuples stored in a table and data stored, as an example, in a B-Tree. In a SciRepo scenario, an Index component fulfils similar requirements. It provides a wide range of indexing structures allowing for combined queries using different indexes.

It supports triggers, through the *Trigger* component, at the primary content resources with a publish/subscribe scheme. This is important to guarantee freshness of information about information objects (up to a certain delay).The primary information objects must undergo a number of transformations to extract the index data, such as color extraction from an image, terms frequency from a textual documents, column names for datasets. This set of features extractors are managed by the *Features Extraction* component that represents the placeholder where plug-and-play extractors can be added and managed.

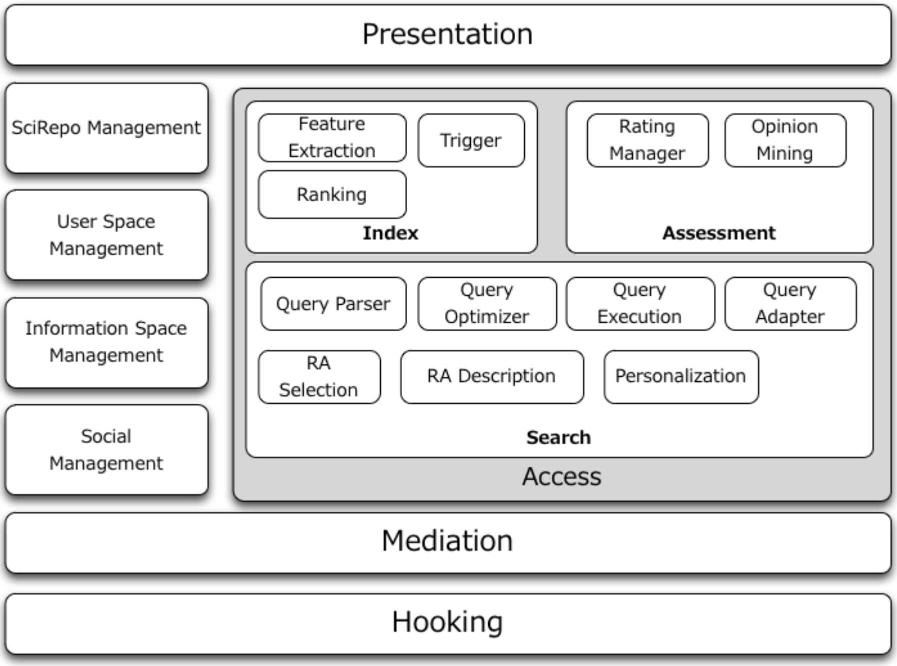


Figure 4.5: The Access Area

Assessment

The Assessment component allows SciRepo users to assess and evaluate the quality of information objects. It is comprised by two main components: the *Rating Manager* component that collects and efficiently store user ratings in terms of percentile of points received, and the *Opinion Mining* component which analyses the contingent text received during the ratings and extracts the opinions it expresses about information objects [47].

Finally, the ratings of the *Rating Manager* could be exploited by the *Ranking module* as a possible factor in the an overall score of each search relevant result.

4.3.5 The User Space Management Area

The management of information about users and groups of users is mainly divided into three functional parts: user management, group management, and role based policy management. The user management part covers functionalities for adding and removing users to/from the SciRepo. The group management part includes use cases to create and remove groups of users as well as to edit the group profiles.

The role based policy management covers the access control to the products. These access controls can prescribe not only who or what process may have access to a specific resource, but also the type of access that is permitted.

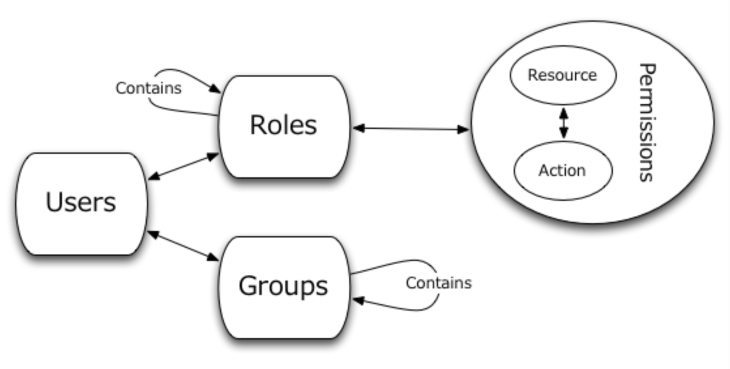


Figure 4.6: Role Based Policy Management

The relationship between the main three area of management is represented in Figure 4.6.

A Role is a job function within the context of an organisation with some associated semantics regarding the authority and responsibility conferred on the user assigned to the role.

A Permission is an approval to perform an operation on one or more RA. Usage rights are modelled as associations between roles, actions and resources. Furthermore, roles are organised in hierarchies allowing a natural way to capture organisational lines of authority and responsibility. Role hierarchies are not constrained to be trees; each role can have several ancestors with the only constraint that cycles are not allowed in the structure.

Even if the groups are introduced to put similar users together or maintain information related to many users, e.g., a contact person, there is no explicit relationship between groups and either permissions or roles. This essentially means that a group is not tailored to grant any right to users. The concept of group and that of role have to be considered orthogonal. Groups are useful to easily manage information related to a set of users, to discover communities with related interests, and to simplify management tasks through common actions on a set of users.

The User Space Management area includes a number of components tailored to provide the required features covering the listed functionalities. A graphical representation of these components is depicted in Figure 4.7.

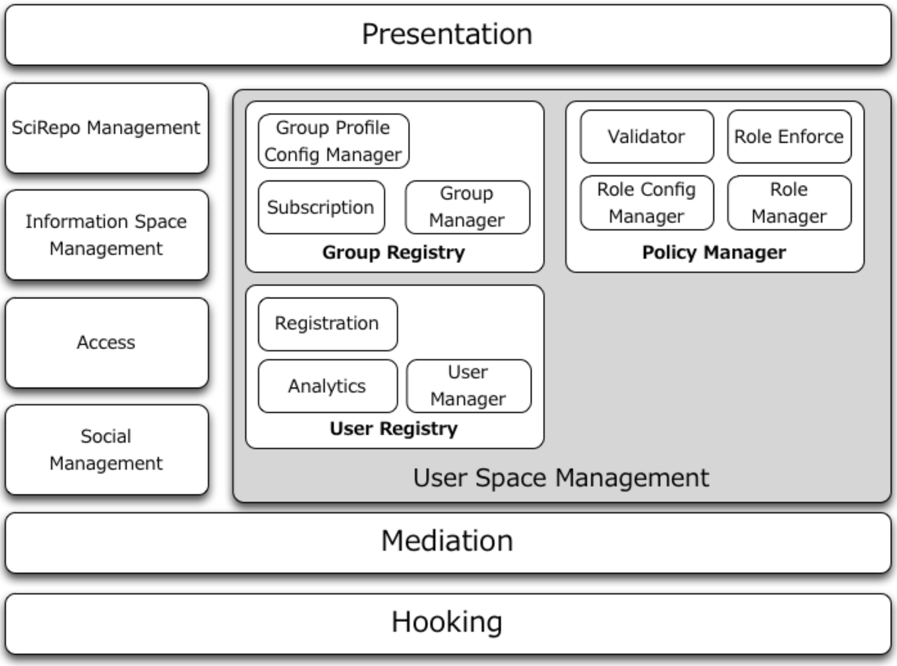


Figure 4.7: The User Space Management area

User Registry

The *User Registry* component provides the mechanisms to manage user profiles compliant with a profile format. These functions include management of the user profile storage; activation of the appropriate mechanisms for their preservation; enrichment of user profiles with third-party applications information (e.g., import of LinkedIn profile); and mechanisms for their discovery and access. In order to perform this wide set of functions, the User Registry exploits the capabilities provided by the three main components described below.

The *Registration* component is responsible for the acquisition of new user profile and for the management of the process that ends with the activation of a new user credential. This process can be executed in one or more steps and can involve one or more actors. The simplest supported process is activated by an end-user that is immediately accepted as new SciRepo user. The standard process also includes confirmation of the user registration through exchange of emails. This mechanism verifies the electronic address of the new user, thus automatically protecting the user against illegal registration threats. The advanced process includes the management of incoming registrations through a human administrator. This means that the registration requests remain pending until explicitly approved by an entitled actor that protects the SciRepo against illegal uses. All processes include the validation of the user profile, its enhancement through the addition of administrative information, and the storage of the user profile through the Storage Manager component described previously.

The *User Manager* component manages the user profiles and assures that they are discoverable and accessible. Its functions include the assignment and/or validation of a unique login, that must be unique not only within the User Registry but also within the Research Infrastructure of which it is a part; the generation of additional Administrative metadata complying with the user profile format; and the transformation of the user profile from how it was submitted into a bytestream that can be stored on suitable hardware through the Profile Registry.

The *Analytics* component manages the user behaviour and feedbacks. This information on may consist in statements recorded from any user or in implicit measures. The former class includes what a user thinks of a Research Product or Research Activity, typically using the Rate or Comment functionalities explained in the End-users perspective of the Reference Model (cf. Sec. 3.2.1). The latter includes measures inferred from available data on user activity, such as products a user has accessed, time spent reading and interacting with the SciRepo.

Group Registry

The *Group Registry* component provides the mechanisms to manage group profiles compliant with a profile format. These functions include management of the group profile storage; activation of the appropriate mechanisms for their preservation; enrichment of group profiles with additional behavioural information; and mechanisms for their discovery and access. Further, as already stated in the Reference Model, implicitly the users belonging to a Research Activity form a Group.

In order to perform this wide set of functions, it exploits the capabilities provided by three main components.

The *Group Profile Config Manager* component defines the appropriate group profile. The group profile is tailored to represent a set of users with specific characteristics that are described in textual forms. It can also indicate one or more moderators that are entitled to manage the group through the management of the user's membership.

The *Subscription* component is responsible for the acquisition of new users in a group. This process can be executed in one or more steps and can involve one or more actors. The simplest supported process is activated by an end-user that is immediately accepted as a new member of the group. The standard process includes the confirmation of the user registration though emails. This process can be activated by the user itself or by a moderator of the group that is entitled to send an invitation request to the user. The advanced process includes the management of incoming subscription through a moderator. This means that the registration requests remain pending until explicitly approved by the moderator that protects the group against illegal participation. All processes include the validation of the group profile and its enhancement through the addition of administrative information.

The *Group Manager* component manages the group profiles and assures that they are discoverable and accessible. Its functions include the assignment and/or validation of unique identifier; and the transformation of the group profile from how it was submitted into a bytestream that can be stored on suitable hardware through the Storage Manager.

Policy Manager

The *Policy Manager* component provides the mechanisms and functions for the definitions of roles, definitions of permissions, associations of users to roles, and retrieval of role based statements.

With role-based access control, access decisions are based on the roles that individual users have as part of a SciRepo. Access rights, expressed through a set of permissions, are grouped by role name, and the use of resources is restricted to individuals authorized to assume the associated role.

The definition of the roles and their configuration with the assignment of permission to operate on information objects is done through the *Role Configuration Manager* component. Role associations can be established when new functionality is provided by the SciRepo, and old functionality can be deleted as organisational functions change and evolve. This simplifies the administration and management of privileges because roles can be updated without updating the privileges for every user on an individual basis.

The user membership into roles can easily be assigned/revoked through the *Role Enforce* component. The process to assign a user to a role is always enforced after verifying that the actor that is invoking this function is entitled and authorised to perform such an operation. This validation of the user credential is done by the *Validator* component that is also able to verify the correctness of the role statement.

Finally, the *Role Manager* component provides the functionality to store, discover, and access role statements.

Profile Registry

The *Profile Registry* component provides the mechanisms and functions for the storage, maintenance, and retrieval of profile manifestations compliant with one of the supported profile formats. These functions include accepting registration of profile formats; receiving profile manifestations; adding them to permanent storage; and managing the level of required replication. It can be configured to exploit its internal storage back-end or to use an external Storage component. The storage of the profiles can also be anticipated by the execution of cryptographic mechanisms to prevent anyone, except the component itself, from reading those data. Many types of data encryption, that represent the basis of network security, can be supported. Common types include Data Encryption Standard and public-key encryption.

4.3.6 The Social Management Area

The set of components belonging to the Social Management area can be divided in four functional parts: *Activity Feed* , *Messaging, Notification* and *Personalization*. The Activity Feed component manages Research Activity feeds. Feeds are used to represent temporally ordered information about the user activities within RAs. Messaging provides end-users with means to participate, interact and contact another user. The Notification component manages a list of happenings organised by date in reverse chronological order related to the users of SciRepo. Finally, all of this can be customisable by means of a *Personalization* component.

The Social Management area graphical representation is depicted in Figure 4.8.

Activity Feed

The *Activity Feed* component provides the mechanisms and functions for the management of end-user social networking activities within a given RA. A Feed is an entry describing the activities of a user in terms of user/verb/object triple where verbs are used as types of activities: to post, to like or to comment a particular information object. Therefore, the Activity Feed depends on the following subcomponents: the *Post Registry*, the *Like Registry* and the *Comment Registry.* Each of these three components must provide functionalities for the efficient storage and retrieval of these type of activities.

Messaging

The *Messaging* component provides the mechanisms and functions for users to interact. There are two components needed to manage the types of interactions identified in the Reference Model. An instant message (IM) component and an email-like

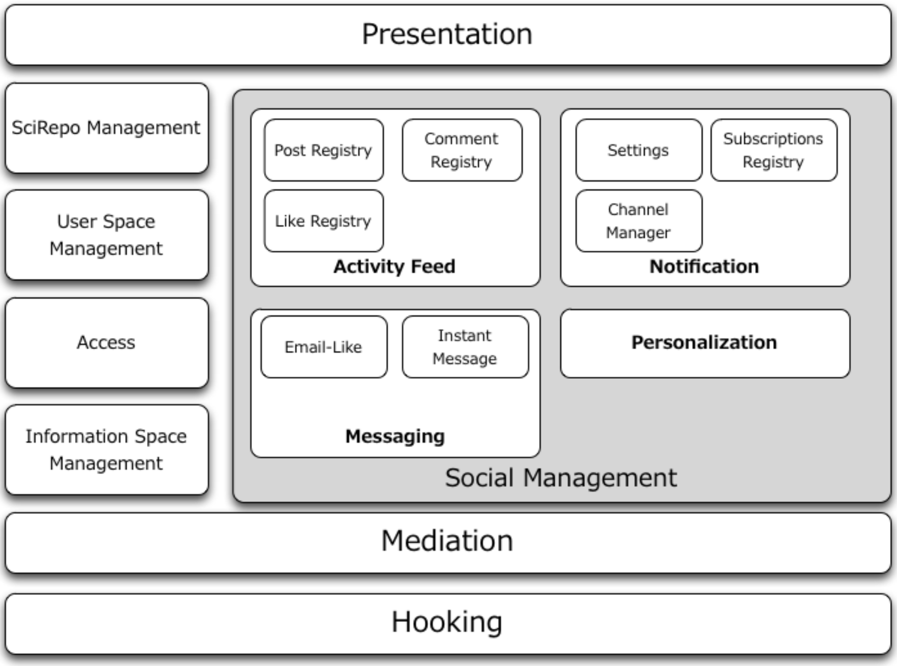


Figure 4.8: The Social Management area

(EML) component. Both components must provide functionalities for the efficient delivery, storage and retrieval of messages. It is important to note that IM and EML feature different requirements in terms of Quality of Service and therefore one storage could be preferable to another, however recently attempts have proved that one single -scalable- storage adoption is possible [17].

Notification

The *Notification* component provides the mechanisms and functions for alerting SciRepo users on an as-it-happens basis. The notifications offer a sense of anticipation and create a productivity boost. Users receive an alert, through a priori selected channel, e.g., email, twitter, notifying them when something of interest has happened in their RAs. For instance, RA members will receive notifications when a complex and time consuming experiment is completed and its results are available, or a task of a collaborative workflow is completed by all the planned participants, or another user mentioned a RP within an ongoing discussion thread and other similar scenarios. There are three subcomponents to take into account: the *Subscription Registry*, which provide functionalities for the efficient storage and retrieval of user subscriptions to RPs or RAs. The *Channel Manager*, containing information about the means through which a user receives notifications. The *Settings* subcomponent, providing functionalities for the customisation on the type of notifications users can receive.

Personalization

The *Personalisation* component provides every user with facilities to customise the overall behaviour of the Social Management area. It enables to specify information including biographic data, interests and skills.

4.3.7 The SciRepo Management Area

The SciRepo management area includes the services needed for the daily maintenance of a SciRepo Platform. This includes management of the system configurations, support to the end-users, management of the access rights, and monitoring of the SciRepo Research Activities.

To support this broad area of functionalities a set of components has been designed as depicted in Figure 4.9.

SciRepo Monitor

The SciRepo Monitoring keeps track of the *Status,* here intended as set of values of all the parameters that define a condition, of the Information Space perceived by SciRepo Administrators (cf. Sec. 3.2.2). Therefore, the status of the building blocks of the SciRepo Platform: *Hosting Nodes,* identifying the hardware devices providing computational and storage capabilities, and *Components,* software packages that may assume the shape of web services, web resources, or modules to deliver a set of related functionalities. SciRepo Monitoring allows supervising the average number of requests managed by the *Component*, the average load of the *Hosting Node*, the average number of queued requests, the latency, the throughput, etc. It does this by relying on the Information System capabilities of the Enabling functional area (cf. Sec. 4.3.9).

User Administration

Access to SciRepo *User Administration* activities is controlled by means of user names and groups. Each User has a unique name and password. Users are placed in functional groups (SciRepo Roles) according to the Research Activities and/or

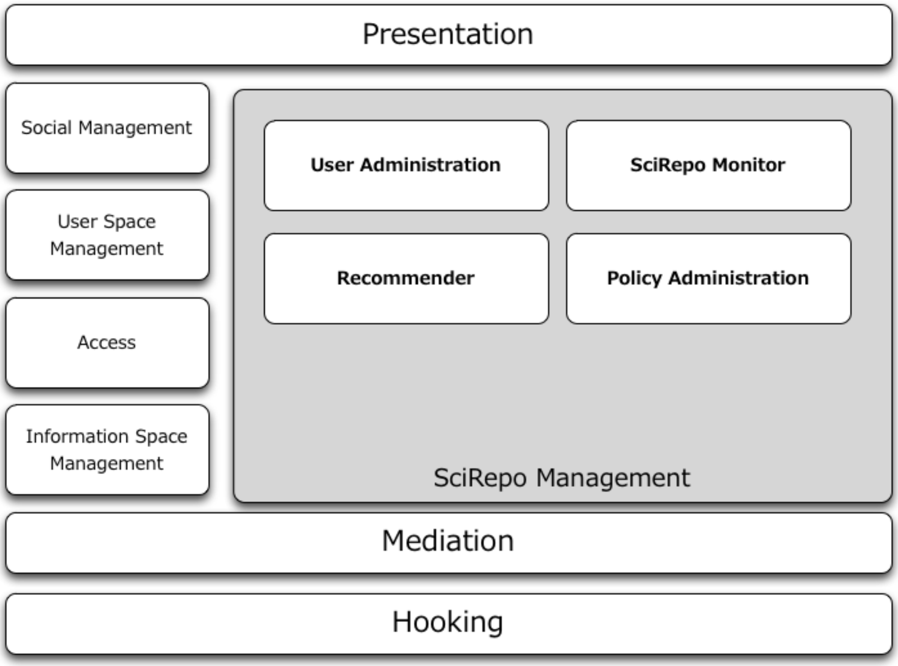


Figure 4.9: The SciRepo Management area

SciRepo features they may use. Information about Users and SciRepo Roles is held by, and is accessible through, this component. When a User Administrator adds or removes users or roles, or views information about existing users or roles, the methods in the component look in the *User Registry* component of the User Space Management area to obtain authorisations to carry out the administrator's request.

Policy Administration

The *Policy Administration* component supports the SciRepo Administrator users providing the interface to define the roles with the associated permissions and to manage them along the SciRepo lifetime; to monitor and manage the user un-registrations; to define the policies for access control to ensure all parties are protected, including authentication of users and disseminated information objects; to define the policies for storage of information objects.

Recommender Component

The *Recommender* component implements the disseminate functionality through which new information objects are advertised to the users whose subscription requests are correlated with their descriptions. For instance this component is in charge of recommending which are the “Trending” Research Activities of a SciRepo, for this it has to consider also the time dimension.

Recommendations can be directly managed by the SciRepo Administrators through the help of the Recommender user interface or can be automatically generated by the component itself. In the former case the Administrator, after analysing any user profile, automatically acquired through the User Administration component, and the list of new information objects, makes her recommendations in accordance with the user subscription requests to RAs or RPs. In the latter case, the Recommender component aims at predicting an individual’s preferences in order to make specific real-time recommendations accordingly. It does this by first learning each individual’s preferences through observing real-time behaviour, i.e., click-through, individual subscription topics, and past behaviour, acquired via the *Analytics* component. The prediction can be based on a content based mechanism, that selects the right information for the right people by comparing representations of content contained in the information objects with representations of information objects that the user is interested in; or on collaborative mechanisms, that allows to work by collecting human judgments for the information objects; identifies users whose information needs and/or tastes are similar to those of the given user and recommends information objects they have liked.

4.3.8 The Presentation area

A SciRepo Platform provides the mechanisms and instruments to build, maintain, and use Research Products. It supports a configurable, distributed, and heterogeneous information-enriched environment providing researchers with collaboration oriented and repository oriented facilities enabling a seamless and complete access to any Research Product and, mostly, the context leading to it. As a result, the usability and accessibility of this environment by different users and systems, with varying needs and capabilities, represent a fundamental property that must be addressed with particular attention. The Presentation area has to include components capable to deliver processes, methodologies, and tools to meet the requirements of end-users, administrators, and developers in a balanced way.

It is important to notice that by interactively exploring Research Activity and Product visualisations uses can obtain overviews, search for trends, make comparisons, and share findings. Thus a SciRepo implements a scenario where researches extract and consult research artifacts, and actively contribute and become the makers of them, also by adding comments, ratings, pursuing modifications, refining, and creating new relations.

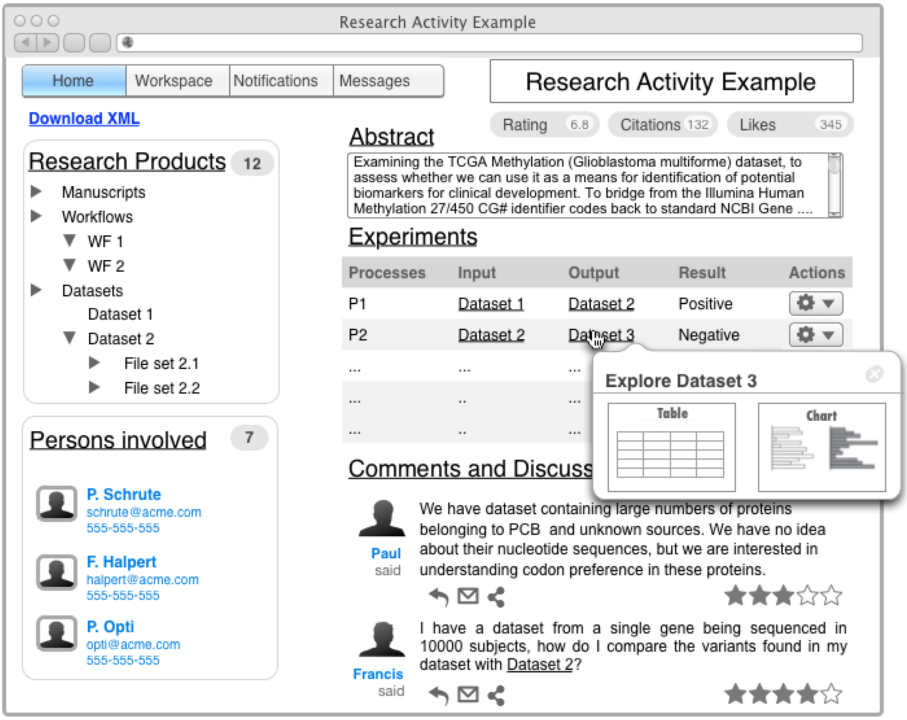
The *User Interface* (UI) component provides lot of functions for SciRepo users, among these: searching, browsing, visualisation of Research Activities, Research Products, User Profiles, and capability for users to traverse their semantic relationships (e.g., to view the RPs associated to a RA, to view User Profiles associated to the ones partaking in a RA etc.). Also possibilities for selection, filtering, and ranking of Information Objects, for social collaboration with peer groups in various contexts, for reusing and re-interpreting information objects in different contexts. All these ways of functioning should be manually or automatically customisable in order to meet the different user types, characteristics and needs.

One of the key feature the UI component has to implement is the ability to display correctly across multiple devices. For instance, mobile devices are often constrained by display size and require a different approach to how content is laid out on screen. There exist a large number of different screen sizes across smartphones, tablets, notebooks, desktops. Screen sizes will always be changing, so it is important that the SciRepo UI would adapt to any size, today or, possibly, in the future. Responsive web design [69] welcomes the needs of the users and the devices they are using. Furthermore, it is advisable that the *User Interface* component supports the following visual interfaces:

- an overview of the Research Activities of a SciRepo a user is involved in. A place where the user is informed on the developments occurring in its community (e.g., Research Activities outcomes);
- an overview of the set of experiments (belonging to a Research Activity) that have run in the Research Infrastructure and the Comments and Discussions related to them;
- an overview of the impact indicators of the Research Activity: the overall Rating the SciRepo gives to this activity, the number of Citations and the Likes received by the other users;
- grouping of Research Products by type (Manuscripts, Workflows, Datasets etc..) and navigable links/specific actions for all the Research Products associated with a Research Activity;
- to visualise the users actually contributing to his/her Research Activity;
- to allow end-user express their position with respect to the ongoing activities;
- to visualise emerging trends and transient patterns, and more generally, visualising knowledge domains[37];

- to provide appropriate visualisation tools to manage different type of data, such as textual manifestations, image data, videos and representations of the semantic relationships occurring in the constituents of any Research Activity. These tools help the users to explore the difference between various versions of the same information object, or help them to navigate through relevant paragraphs, or help to handle diverse and large scale multimedia and mixed-media data;
- to visualise user interactions with data in relation to available Information Objects, in order to improve the SciRepo use.

An example of the web page of a Research Activity is given in Fig. 4.10. This page contains links and actions for all the research products associated with it on the reader's left part. In this specific example, we can see twelve Research Products grouped together by type (Manuscripts, Workflows, Datasets etc..); At the bottom of this part the people actually contributing to the Research Activity are shown.



Abstract

Examining the TCGA Methylation (Glioblastoma multiforme) dataset, to assess whether we can use it as a means for identification of potential biomarkers for clinical development. To bridge from the Illumina Human Methylation 27/450 CG# identifier codes back to standard NCBI Gene ....

Experiments

Processes	Input	Output	Result	Actions
P1	Dataset 1	Dataset 2	Positive	⚙ ▾
P2	Dataset 2	Dataset 3	Negative	⚙ ▾
...	...	...		
...	..	...		
...	..	...		

Comments and Discuss

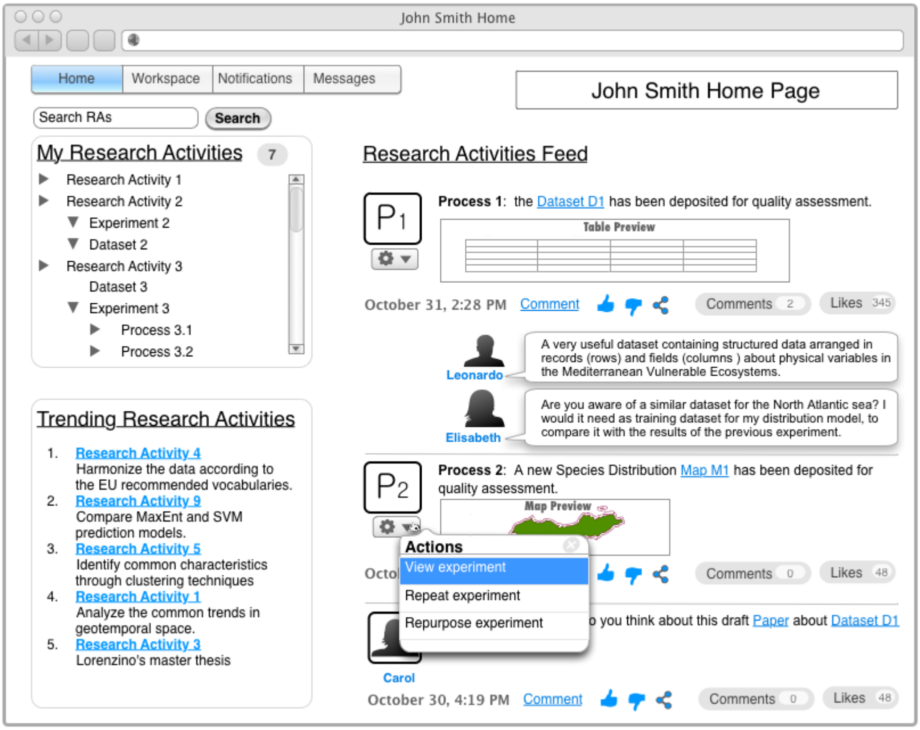
Paul said: We have dataset containing large numbers of proteins belonging to PCB and unknown sources. We have no idea about their nucleotide sequences, but we are interested in understanding codon preference in these proteins. ★★★☆☆

Francis said: I have a dataset from a single gene being sequenced in 10000 subjects, how do I compare the variants found in my dataset with Dataset 2? ★★★★☆

Figure 4.10: An Example of a SciRepo Research Activity Web Page

On the reader's right part instead we can see that a Research Activity is characterised by a description (Abstract), the set of experiments that have run in the "source" Research Infrastructure and the Comments and Discussions related to the Research Activity. The page also shows that a Research Activity is characterised by a series of impact indicators resulting from the "use" of the activity as a whole, for instance the overall *Rating* the SciRepo gives to this activity, the number of *Citations* and the *Likes* received by the other users.

An example of the personal web page every SciRepo user is provided with is in Fig. 4.11. This is a sort of console where the user is acquainted with the happenings occurring in its community (e.g., Research Activities outcomes) as well as have easy access to the Research Activities he / she is involved in. Through it the user can express his / her position with respect to the ongoing activities.



1. Research Activity 4
   Harmonize the data according to the EU recommended vocabularies.
2. Research Activity 9
   Compare MaxEnt and SVM prediction models.
3. Research Activity 5
   Identify common characteristics through clustering techniques
4. Research Activity 1
   Analyze the common trends in geotemporal space.
5. Research Activity 3
   Lorenzino's master thesis

Process 1: the Dataset D1 has been deposited for quality assessment.

October 31, 2:28 PM

Leonardo: A very useful dataset containing structured data arranged in records (rows) and fields (columns) about physical variables in the Mediterranean Vulnerable Ecosystems.

Elisabeth: Are you aware of a similar dataset for the North Atlantic sea? I would it need as training dataset for my distribution model, to compare it with the results of the previous experiment.

Process 2: A new Species Distribution Map M1 has been deposited for quality assessment.

o you think about this draft Paper about Dataset D1

October 30, 4:19 PM

Figure 4.11: An Example of a SciRepo End-user Web Page

The Presentation area does not only provide a human-based application to access and manage the contents of a SciRepo. It should also include components that support and promote the export of SciRepo Research Products to other third-party applications.

This is the case of the Export API, for instance an *OAI-PMH Publisher* component exposes and makes available the descriptive metadata of the managed information objects through the Open Archives Initiative Protocol for Metadata Harvesting or the *OAI-ORE Publisher* (Open Archives Initiative Object Reuse and Exchange) that makes available standards for the description and exchange of aggregations of compound digital objects.

However it is envisaged also to provide a custom API Endpoint too, exploitable through different protocols (e.g. JSON, REST) for better customising the mapping with the SciRepo Data Model. In fact, the standard OAI-PMH publisher acts as an information space virtualiser having the ability to expose the information objects organised in sets that may not correspond to the managed products.

4.3.9 The Enabling area

As we have seen so far, there is a set of core SciRepo functionalities, such as search, retrieval, and access to information objects, that any SciRepo Platform supports. Nevertheless, there are a set of requirements that characterise any SciRepo application and that make difficult both the realisation and maintainability of the software, and the provision of an operational SciRepo service to the end-user communities. More precisely one should take into account that:

- each SciRepo Platform must provide, in addition to the core functionalities, other specific functionalities for serving application-specific requirements;
- during the SciRepo lifetime new organisations may join the SciRepo by bringing their content and additional functionalities may be required to satisfy new needs. Therefore, a SciRepo Platform must be able to dynamically evolve by adapting itself to these new situations;
- the handling of a SciRepo can be expensive in terms of financial, infrastructure and human resources. The adoption of a SciRepo federated model could be as a solution to this problem. By following this model, multiple organisations can set up a SciRepo by sharing their resources according to their own policies. For example, they can decide to share and distribute the social networking services but store their information objects locally.

In order to satisfy these requirements, we have proposed to rely on a component-based architecture (an architecture in which all the functionalities are provided by independent components with well-defined interfaces). This organisation, where components are enabled to expose their interface and service consumers are entitled to find the more appropriate ones, provides the necessary conditions for supporting federation, openness and dynamic evolution. Despite that, the component-based architecture opens to a number of issues that have to be taken into account carefully to avoid to compromise the sustainability of the SciRepo Platform. These issues are related with the following observations:

- there are basic functionalities that are needed for creating and operating a SciRepo Platform. These functionalities are global in nature, i.e. they are not related with a particular service. In particular, they support the gathering, storage, and publishing of information about components in order to supply the necessary integration and mediation among them;
- there are components, such as the *Search*, the *Deposition* etc., whose logic is mainly based on the composition of other component functionalities. General purpose mechanisms for combining the underlying components in order to form a complex flow of communication among them and thus enable the construction of “new functionalities” is envisaged as a solution for reducing the component programming complexity;
- an additional set of functionalities aimed at ensuring the desired quality of service should be provided by a SciRepo Platform in order to manage the set of dynamic, customisable and independent components described so far. This set includes many functions related to the security, e.g., authorization of the request, encryption and decryption as required, validation, etc.; and dynamic rerouting for failover or load balancing.

In order to satisfy these architectural requirements, we designed the SciRepo Reference architecture equipped with an *Enabling area* that includes a number of components as graphically depicted in Figure 4.1.

Authentication and Authorization

The *Authentication* and *Authorization* components provide the necessary support for the management of the user credentials and the validation of the accessing rights.

Specifically, the Authentication deals with user identity and credentials management issues. It provides the mechanisms for the management of the security based on the simple login and password model, on the more secure one time password, and on the sophisticated Public Key Infrastructure (PKI) with the use of X.509 End Entity Certificates (EEC) released by a Certification Authority (CA). It also includes support for delegation and single sign on by exploiting secure delegated access on behalf of a resource owner via open standard frameworks such as OAuth[1], OAuth2 or OpenID[2].

If the X.509 Proxy Certificates standard is adopted the authorization information can be managed as certificate extensions. These extensions are included in the PC during its creation. This functionality allows distributing the Authorization support at any hosting node to resolve locally authorization issues, without the need to contact a remote authorization component. Moreover, this model bounds the lifetime and the extent of authorisations to the information contained in the Proxy Certificate.

Accounting

The *Accounting* component provides the necessary support for the measurements of consumptions of specific functionalities. It deals with the system's logging capabilities: who did what, where, and when, for how long, or for how much. Accounting is not only meant for emergency use, such as a system breach, but is also useful for the validation of continuous operations, verifying that components behave as expected.

Information Service

As in any component-based environment, the purpose of the *Information Service* (IS) component is to allow other components to be aware of the environment they operate in. It maintains the most up to date information about the set of available hosting nodes that compose infrastructure together with the status of the SciRepo components. In particular, the IS provides mechanisms for:

- gathering, storing and supplying information about the components operating in the SciRepo;
- monitoring the components configuration and status information.

Besides, it allows users and other components to discover what resources are deployed in the SciRepo and to monitor those resources; it provides publish and subscribe mechanisms, that provide support for an event-driven model in which an event that occurs in a component can trigger an action in another one that is registered for that notification.; it provides itself a trigger interface that can be configured to take action when pre-configured conditions are met; and finally, it archives information to allow historical query execution.

[1] Oauth: an open protocol to allow secure authorization in a simple and standard method from web, mobile and desktop applications - http://oauth.net

[2] OpenId Foundation website: http://openid.net

Broker

The Broker component enables the integration of the various components through the introduction of a reliable set of capabilities. These capabilities include a message transformation mechanism, that transforms data into a common data format that is understandable by both the sending and the receiving components; and an intelligent routing functionality, that frees the sending component from having knowledge of the location of the component a message is direct to.

In order to perform his job, the Broker performs a matching between the requirements expressed in the request and his knowledge about components and hosting nodes. However, it does not gather by itself the status of the whole hosting nodes environment; rather, it relies its computations on information supplied by the *Information Service* component. This status information, periodically fetched from the IS or notified by it upon any change in the hosting node status, is stored in the Broker Catalogue.

It is important to notice that exploiting the features of this component, rerouting for fail over and load balancing can easily be supported without increasing the complexity of the SciRepo components logic or sharing to all the SciRepo components the semantic of the knowledge infrastructure status and of its management.

4.3.10 The Application Framework

The component-based architecture presented so far has proved to provide the following benefits:

- *Reuse*, i.e. the ability to create software code that is reusable in multiple applications;
- *Efficiency*, i.e. ability to quickly and easily create new applications using a combination of new and old components, along with the ability to focus on the data to be shared rather than the implementation underneath;
- *Loose technology coupling*, i.e. the ability to model software code independently of the infrastructural environment and exploit message exchanges to govern the component-to-component cooperations;

If all applications were to use a common programming interface and interoperability protocol, however, the job of IT would be much simpler, complexity would be reduced, and existing functionality could be more easily reused. After a common programming interface is in place, through which any application can be accessed, existing IT infrastructure can be more easily replaced and modernised.

This is where Service Oriented Architecture (SOA) steps in. SOA is an architectural paradigm for components of a system and interactions or patterns between them. It is an evolution of the component-based architecture because besides the breakdown of the application in components, their description, advertising and discovery, it adds the specification of an associated data model and the separation of the interface from the implementation logic [45]. The idea of separating an interface from its implementation to create a software service definition has been well proven in J2EE, and CORBA before that. But the ability to more cleanly and completely separate, basically by interpreting a text file, a functionality description from its execution environment is new.

Two standards, whose universal adoption proved their intrinsic value, represent the key technologies for a SOA implementation. These are Web Services (WS) and XML, in pair with its “lightweight” alternative JSON[3] as data-interchange format.

XML is a common, independent data format that provides:

- Standard data types and structures, independent of any programming language, development environment, or software system;
- Pervasive technology for defining business documents and exchanging business information, including standard vocabularies for many industries;
- Ubiquitous software for handling operations on XML, including parsers, queries, and transformations.

Web services are XML-based technologies for messaging (or JSON-based), service description, discovery, and extended features, providing:

- Pervasive, open standards for distributed computing interface descriptions and document exchange via messages;
- Independence from the underlying execution technology and application platforms;
- Extensibility for enterprise qualities of service such as security, reliability, and transactions;
- Support for composite applications such as business process flows, multi-channel access, and rapid integration.

A Web service is a software system designed to support interoperable machine-to-machine message exchange over a network. It has an interface described in a machine-processable format, in accordance with the Web Services Definition Language (WSDL). Other systems interact with the Web service in a way prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML/JSON serialisation in conjunction with other Web-related standards. A WS is therefore defined in terms of the message exchange patterns (MEPs) it supports, such as request/response, one-way asynchronous, or publish/subscribe.

[3] JavaScript Object Notation (JSON): http://www.json.org/

A schema for the data contained in the message is used as the main part of the description, also known with the term contract, used by a service requester to use a service provider. Other items of metadata describe the network address for the service, the operations it supports, and its requirements for reliability, security, and transactionality.

As a consequence, developing a service is different from developing an object because a service is defined by the messages it exchanges with other services, rather than a method signature. A service must be defined at a higher level of abstraction than an object because it is possible to map a service definition to a procedure-oriented language, or to a message queuing system such as JMS or MSMQ, as well as to an object-oriented system such as J2EE or the .NET Framework.

An important part of the definition of a service is that its description is separated from its executable agent. One description might have multiple different executable agents associated with it. Similarly, one agent might support multiple descriptions. The description is separated from the execution environment using a mapping layer (sometimes also called a transformation layer). The mapping layer is often implemented using proxies and stubs. The mapping layer is responsible for accepting the message, transforming the XML data to the native format, and dispatching the data to the executable agent.

The presented SciRepo components can be implemented as web services or as software libraries that are then used by other web services. Usually, the use of reusable code libraries or class libraries to implement common functions that are loaded or linked into applications is associated to a pre-service oriented development environment. Instead in SOA-based applications, common functions, as well as typical system functions such as security checks, transaction coordination, and auditing are implemented using services. However, the performance implications of accessing services instead of using internal functions must be assessed because using a service typically consumes more computing and networking resources than reusable code libraries. The key for a successful SciRepo Platform implementation based on SOA is therefore strongly influenced by the correct design and function of the reusable software libraries in web services based applications.

The envisaged Enabling area is instead strongly influenced by the adoption of the WS framework along the security aspects, the information storage and discovery, and the process orchestration.

Security Aspect

Threats to Web services involve threats to the hosting node system, the application and the entire network infrastructure. To secure Web services, a range of XML-based security mechanisms are needed to solve problems related to authentication, role-based access control, distributed security policy enforcement, message layer security that accommodate the presence of intermediaries.

There are not yet broadly-adopted specifications for Web services security. As a result developers can either build up services that do not use these capabilities or can develop ad-hoc solutions that may lead to interoperability problems.

Web services implementations may require point-to-point and/or end-to-end security mechanisms, depending upon the degree of threat or risk. Traditional, connection-oriented, point-to-point security mechanisms, such as Transport Layer Security (SS-L/TLS), Virtual Private Networks (VPNs), IPSec (Internet Protocol Security), and Secure Multipurpose Internet Mail Exchange (S/MIME), may not meet the end-to-end security requirements of Web services.

In general, Web services use a message-based approach that enables complex interactions that can include the routing of messages between and across various trust domains. A message might travel between various intermediaries before it reaches its destination. Therefore, message-level security is important as opposed to point-to-point, transport-level, security.

Secure Messaging ensures privacy, confidentiality and integrity of interactions. Techniques that ensure channel security can be used for securing messages, but only in a few limited cases. Examples include a static direct connection between a requester agent and a provider agent, that can be appropriate in some component-to-component connection illustrated in the previous sections. However, in the general case, message security techniques such as encryption and signing of the message payload can be used in routing and reliable messaging.

It is evident therefore that there is more than one solution and that security is a balance of assessed risk and cost of countermeasures. Depending on the application risk tolerance, point-to-point transport level security can provide enough security countermeasures.

Information Service and Discovery Aspect

The Information Service is also strongly influenced by the WS framework that gives three alternatives to implement it: the Registry, Index, and Peer-to-Peer (P2P) approaches.

The Registry approach is based on an authoritative, centrally controlled service that stores information. Publishing a service description requires an active step by the provider entity that explicitly places the information into the registry. The Universal Description, Discovery, and Integration (UDDI) platform is often seen as an example of the registry approach, but it can also be used as an index.

The Index Approach is based on a distributed and replicated network of services that store information. Publishing a service description becomes a passive step because the provider entity exposes the service and functional descriptions on the Web, and the indexes collect them without the provider entity's specific knowledge.

In this case different indexes could provide different kinds of information, some richer, some sparser. Again, UDDI could be used as a means to implement an individual index.

Peer-to-Peer (P2P) computing provides an alternative that does not rely on centralised or distributed registries. Rather it allows Web services to discover each other dynamically. Under this view, a Web service is a node in a network of peers. At discovery time, a requester agent queries its neighbours in search of a suitable Web service. If any one of them matches the request, then it replies. Otherwise each query is propagated through the network until a particular termination criterion is reached. With this approach nodes contact each other directly, so the information they receive is known to be current. In contrast, in the registry or index approach there may be significant latency between the time a Web service is updated and the updated description is reflected in the registry or index.

The reliability provided by the high connectivity of P2P systems comes with performance costs and lack of guarantees of predicting the path of propagation. Any node in the P2P network has to provide the resources needed to guarantee query propagations and response routing, which in turn means that most of the time the node acts as a relayer of information that may be of no interest to the node itself. This results in inefficiencies and large overhead especially as the nodes become more numerous and connectivity increases. Furthermore, there may be no guarantee that a request will spread across the entire network, therefore there is no guarantee to find the providers of a service.

Process Orchestration Aspect

Rather than have Web services invoking each other using one or more of the message exchange patterns supported by SOAP and WSDL, the adoption of the WS framework gives the foundation for a systematically use of complex interaction patterns in long-running business process flows with exception handling, branching, and parallel execution.

To accomplish this, the Process Engine has to preserve context and provide correlation mechanisms across multiple services. A Web service orchestration may also be published as a Web service, providing an interface that encapsulates a sequence of other Web services. Entire application suites can be built up at multiple levels of encapsulation, from those that encapsulate a single software module to those that encapsulate a complex flow of other Web services.

Industry has reached a consensus around a single orchestration specification: the OASIS Web Services Business Process Execution Language (WS-BPEL) and its successor, WS-BPEL 2.0 was ratified as a standard in 2007. As an execution language, WS-BPEL defines how to represent the activities in a business process, along with flow control logic, data, message correlation, exception handling, and more. WS-BPEL assumes that Web services are defined using WSDL and policy assertions that identify any extended features. Typically, a flow is initiated by the arrival of an XML document, and so the document-oriented Web services style tends to be used for modelling the entry point to a flow. Parts of the document are then typically extracted and operated upon by the individual tasks in the flow.

The WS-BPEL specification differs from other extended specifications in that it defines an executable language compatible with various software systems that drive business process automation. Whereas most other Web services specifications are XML representations of existing distributed computing features and capabilities that extend SOAP headers, orchestration represents the requirement for composing Web services in a declarative manner. In the case of WS-BPEL micro-flows, an alternative is certainly represented by the use of mediation flows, provided within the Enterprise Service Bus pattern (ESB) [89], to call Web services. However, when the process orchestration calls for a process with complex logic, as explained in the previous, WS-BPEL has container activities, such as while loops and scopes that ESB does not support. The logic in an ESB is quite basic, while WS-BPEL can handle more complex cases.

5

Conclusions

The initiatives aiming at enlarging and strengthening scientific communication therefore to meet the expectations of modern science are hindered by two major factors: cultural barriers (e.g., lack of reward, additional effort) and methodological barriers (e.g., many repositories to deal with).

This fact has motivated the work presented in this dissertation, we first outlined the problem space and investigated the current practices by performing an analysis of the Research Publishing domain and a Survey on the state of the art offering Scientific Data Repositories provide. This Survey collects the practices and approaches for data publishing promoted by generalist repositories, i.e., repositories accepting the publication of any dataset typology, and offers a short guide to the steps scientists can take to ensure that their data and associated analyses continue to be of value and to be recognised.

We reported a set of drawbacks affecting current publishing practices and discussed how they limit the effective interpretation of research results, their correct evaluation and reuse, eventually reducing the number of products eligible for publishing.

This study has helped us in identifying and introducing the notion of Science 2.0 Repository (SciRepo), aiming at overcoming the methodological barriers by providing scientists with an integrated and innovative environment that supports “within” and “during” scholarly communication workflows (deposition, quality assessment and dissemination). In contrast with the identified drawbacks we explained how scholarly communication workflows are realised by SciRepo, discussing the benefits of research publishing exploiting it along the three publishing phases: during *i)* the deposition phase Research Products remain *in context*, and *“alive”* with no *extra-cost* and *support to alternative products*, during *ii)* the quality assessment phase, the evaluation of Research Products is *continuous and in context* and the RI community has the ability to perform *self-assessment* over them, also by profiting of social-like tools or discussion threads, and *iii)* during the dissemination phase, this gets *unified* i.e., scientists are offered with marketplace facilities to discover and access Research Products that subsume those typically offered by publication and dataset repositories, and consequently *automatic and complete* as Research Product authors are less burdened by routine-boredom activities of metadata information and relationship curation.

We described our vision of this new type of repository, expected to be offered as a platform that every Research Infrastructure can use, configure and deploy to extend the working environment of its community, clarifying that SciRepo is conceived to nicely integrate and complement the offering of Research Infrastructures towards holistic scholarly communication practices. To support such a paradigm a common understanding of what a SciRepo is and what its characteristics are must be shared by all the actors interacting with it.

Therefore we transformed the initial intuitive notion into a Reference Model for a SciRepo platform, an abstract work intended to comprehend significant concepts and relationships. These are expressed through concept maps and across three different perspectives (highlighting the needs of the different actors that operate with SciRepo). We identified 83 distinct concepts and 32 distinct relationships that proved to be helpful in providing a common understanding of the platform and semantics that can be used unambiguously across and between different implementations.

To complement this work we also designed a Reference Architecture for a SciRepo platform, a template solution that maps the functionalities defined in the Reference Model onto software components implementing them. One of the main challenges addressed was to make this inherently abstract Reference Architecture understandable by providing sufficient specific information and guidelines.

Besides the functionalities expressed in the Reference Model, we identified 8 important quality attributes for a SciRepo platform and presented them in the form of non functional requirements. These attributes place restrictions on the platform being developed, its development process, and specify external constraints that any SciRepo platform must meet. Then, we identified a set of architecture principles and best practices, and designed an architecture blueprint standing at the base of any implementation of a general purpose SciRepo platform. The Reference Architecture comprises 10 functional areas and is composed by 56 different software components (and subcomponents) that can be helpful in deriving a concrete architecture, thus facilitating the realisation of a SciRepo over any ICT-based research infrastructure environment with limited costs and efforts if compared with from-scratch approaches.

Future steps in this direction will be to define a Reference Implementation and device a general purpose software toolkit standing at the base of the implementation of a SciRepo platform, part of this work was implemented in the context of the D4Science infrastructure and proved its benefits [7] [6].

6

List of publications

This dissertation presents a part of the work carried out during the time of this doctorate at the University of Pisa, Department of Information Engineering. In the following we add, as references, the list of the author's publications that helped in conceiving the innovations presented here, ordered by type in chronological reverse order:

Journal articles

- Assante M., Candela L., Castelli D., Manghi P., Pagano P. *Science 2.0 Repositories: Time for a Change in Scholarly Communication*. In: D-Lib Magazine, vol. 21 (1/2) article n. 4. Special issue on Linking and Contextualizing Publications and Datasets. Laurence Lannom (ed.). doi:10.1045/january2015-assante
- Assante M., Candela L., Castelli D., Mangiacrapa F., Pagano P. *A social networking research environment for scientific data sharing: the D4Science offering*. In: The Grey Journal (TGJ): an international journal on grey literature, vol. 10 (2) pp. 151 - 158. D.J. Farace, J. Frantzen (eds.). TextRelease, 2014.
- Assante M., Candela L., Pagano P. *An environment supporting the production of live research objects*. In: The Grey Journal (TGJ): An international journal on grey literature, vol. 9 (1) pp. 24 - 31. D.J. Farace; J. Frantzen (ed.). TextRelease, Amsterdam, 2013.

Conference Proceedings

- Assante M., Candela L., Castelli D., Gioia A., Mangiacrapa F., Pagano P. *A social networking research environment for scientific data sharing: the D4Science offering*. In: GL15 - Fifteenth International Conference on Grey Literature. The Grey Audit: a Field Assessment in Grey Literature (Bratislava, Slovakia, 2-3 December

2013). Proceedings, vol. 15 pp. 151 - 158. D.J. Farace, J. Frantzen, GreyNet InternationalGrey Literature Network Service (eds.). (GL Conference Series, ISSN 1385 2316, vol. 15). TextRelease, Amsterdam, 2014.

- Assante M., Candela L., Pagano P. *An environment supporting the production of live research objects*. In: GL14 - Fourteenth International Conference on Grey Literature. Tracking Innovation through Grey Literature (Rome, 29-30 November 2012). Proceedings, pp. 79 - 86. D. Farace, J. Frantzen, GreyNet, GreyNet, Grey Literature Network Service (eds.). (GL-conference series, ISSN 1385-2308, vol. 14). TextRelease, Amsterdam, 2013.
- Assante M., Pagano P., Candela L., De Faveri F., Lelii L. *An approach to virtual research environment user interfaces dynamic construction*. In: TPDL 2011 - Research and Advanced Technology for Digital Libraries. 15th International Conference on Theory and Practice of Digital Libraries (Berlin/Heidelberg, 26-28 September 2011). Proceedings, pp. 101 - 109. Gradmann Stefan, Borri Francesca, Meghini Carlo, Schuldt Heiko (eds.). (Lecture Notes in Computer Science, vol. 6966). Springer, 2011.

Other Publications

- Assante M., Candela L., Manghi P., Pagano P., Castelli D. *Providing research infrastructures with data publishing*. In: ERCIM News, vol. 100 pp. 20 - 21. Special theme: Scientific Data Sharing and Reuse. ERCIM, 2015.
- Assante M., Candela L., Castelli D., Manghi P., Pagano P. *Research Infrastructure Scientific Communication Systems*. Technical report, 2014.
- Assante M., Candela L., Castelli D., Pagano P. *The D4Science research-oriented social networking facilities.* In: ERCIM News, vol. 96 (January 2014) article n. 30. ERCIM, 2014.
- Fabriani P., Giammatteo G., Assante M., Laskaris N. iMARINE FP7 Research Project - *Software release activity report. Data e-Infrastructure Initiative for Fisheries Management and Conservation of Marine Living Resources*. Deliverable D7.4, 2014.
- Assante M., Candela L., Castelli D., Mangiacrapa F., Pagano P. *The D4Science social networking facilities*. Technical report, 2013.
- Castelli D., Assante M., Candela L., De Faveri F., Pagano P. *Ambienti virtuali di supporto alla ricerca = Virtual Research Environments*. In: Le tecnologie del CNR per il mare / Marine Technologies. pp. 145 - 145. Marco Faimail (ed.). Roma: Consiglio Nazionale delle Ricerche

References

1. Euan Adie and William Roe. Altmetric: enriching scholarly content with article-level discussion and metrics. *Learned Publishing*, 26(1):11–17, 2013.
2. Liz Allen, Jo Scott, Amy Brand, Marjorie Hlava, and Micah Altman. Publishing: Credit where credit is due. *Nature*, pages 312–313, April 2014.
3. Micah Altman and Gary King. A Proposed Standard for the Scholarly Citation of Quantitative Data. *D-Lib Magazine*, 13(3), 2008.
4. Patrick Andreoli Versbach and Frank Mueller-Langer. Open Access to Data: An Ideal Professed but Not Practised. *Social Science Research Network Working Paper Series*, February 2013.
5. Andrew Asher, Kiyomi Deards, Maria Esteva, Martin Halbert, Lori Jahnke, Chris Jordan, Spencer D. C. Keralis, Sivakumar (Siva) Kulasekaran, William E. Moen, Shannon Stark, Tomislav Urban, and David Walling. Research data management: Principles, practices, and prospects. Technical report, Council on Library and Information Resources, 2013.
6. M. Assante, L. Candela, D. Castelli, and P. Pagano. Social Networking Research Environment for Scientific Data Sharing: The D4Science Offering. *The Grey Journal*, 10(2):65–71, 2014.
7. M. Assante, L. Candela, and P. Pagano. An environment supporting the production of live research objects. *Grey Journal*, 9(1):24–31, 2013.
8. Massimiliano Assante, Leonardo Candela, Donatella Castelli, Paolo Manghi, and Pasquale Pagano. Science 2.0 repositories: Time for a change in scholarly communication. *D-Lib Magazine*, 21(1/2), 2015.

9. Massimiliano Assante, Leonardo Candela, Donatella Castelli, and Pasquale Pagano. The D4Science Research-Oriented Social Networking Facilities. *ERCIM News*, 2014(96), 2014.
10. Massimiliano Assante, Pasquale Pagano, Leonardo Candela, Federico De Faveri, and Lucio Lelii. An Approach to Virtual Research Environment User Interfaces Dynamic Construction. In Stefan Gradmann, Francesca Borri, Carlo Meghini, and Heiko Schuldt, editors, *Research and Advanced Technology for Digital Libraries*, volume 6966 of *Lecture Notes in Computer Science*, pages 101–109. Springer Berlin Heidelberg, 2011.
11. T. K. Attwood, D. B. Kell, P. McDermott, J. Marsh, S. R. Pettifer, and D. Thorne. Utopia documents: linking scholarly literature with research data. *Bioinformatics*, 26(18):i568–i574, September 2010.
12. Alexander Ball, Sean Chen, Jane Greenberg, Cristina Perez, Keith Jeffery, and Rebecca Koskela. Building a disciplinary metadata standards directory. *International Journal of Digital Curation*, 9(1):142–151, 2014.
13. Chen Ball, A. J., S Greenberg, C. J., Perez, K. Jeffery, and R. Koskela. Building a disciplinary metadata standards directory. *International Journal of Digital Curation*, 9(1):142–151, 2014.
14. Alessia Bardi and Paolo Manghi. Enhanced publications: Data models and information systems. *LIBER Quarterly*, 22(0), 2014.
15. Alessia Bardi and Paolo Manghi. A Framework Supporting the Shift from Traditional Digital Publications to Enhanced Publications. *D-Lib Magazine*, 21(1/2), January 2015.
16. Sönke Bartling and Sascha Friesike. Towards another scientific revolution. In *Opening Science*, pages 3–15. Springer International Publishing, 2014.
17. Mikhail Bautin, Guoqiang J. Chen, Pritam Damania, Prakash Khemani, Karthik Ranganathan, Nicolas Spiegelberg, Liyin Tang, Madhuwanti Vaidya, and Facebook Inc. Storage Infrastructure Behind Facebook Messages Using HBase at Scale.
18. S. Bechhofer, D. De Roure, M. Gamble, C. Goble, and I. Buchan. Research objects: Towards exchange and reuse of digital knowledge. *Nature Precedings <http://dx.doi.org/10.1038/npre.2010.4626.1>*, 2010.
19. Francine Berman. Got data? a guide to data preservation in the information age. *Communications of the ACM*, 51(12):50–56, 2008.

20. J. Bobadilla, F. Ortega, A. Hernando, and A. GutiéRrez. Recommender systems survey. *Know.-Based Syst.*, 46:109–132, July 2013.

21. Christine L. Borgman. The Conundrum of Sharing Research Data. *Social Science Research Network Working Paper Series*, June 2011.

22. Christine L. Borgman. *Big Data, Little Data, No Data: Scholarship in the Networked World*. The MIT Press, 2015.

23. Philip E. Bourne, Tim Clark, Robert Dale, Anita de Waard, Ivan Herman, Eduard H. Hovy, and David Shotton. Improving the future of research communication and e-scholarship. Force11 white paper, Force11, 2012.

24. Philip E. Bourne, Timothy W. Clark, Robert Dale, Anita de Waard, Ivan Herman, Eduard H. Hovy, and David Shotton. Improving The Future of Research Communications and e-Scholarship (Dagstuhl Perspectives Workshop 11331). *Dagstuhl Manifestos*, 1(1):41–60, 2012.

25. Daan Broeder and Laurence Lannom. Data type registries: A research data alliance working group. *D-Lib Magazine*, 20(1/2), 2014.

26. Catherine Jones Bryan Lawrence, Brian Matthews, Sam Pepler, and Sarah Callaghan. Citation and peer review of data: Moving towards formal data publication. *International Journal of Digital Curation*, 6(2):4–37, 2011.

27. Daniel Burda and Frank Teuteberg. Sustaining accessibility of information through digital preservation: A literature review. *Journal of Information Science*, 39(4):442–458, 2013.

28. Frank Buschmann, Kevin Henney, and Douglas C. Schmidt. *Pattern-Oriented Software Architecture, Volume 4, A Pattern Language for Distributed Computing*. Wiley, June 2007.

29. Frank Buschmann, Kevlin Henney, and Douglas Schmidt. *Pattern Oriented Software Architecture: On Patterns and Pattern Languages (Wiley Software Patterns Series)*. John Wiley & Sons, 2007.

30. Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, and Michael Stal. *Pattern-oriented Software Architecture: A System of Patterns*. John Wiley &amp; Sons, Inc., New York, NY, USA, 1996.

31. Sarah Callaghan, Steve Donegan, Sam Pepler, Mark Thorley, Nathan Cunningham, Peter Kirsch, Linda Ault, Patrick Bell, Rod Bowie, Adam Leadbetter, Roy Lowry, Gwen Moncoiffé, Kate Harrison, Ben Smith-Haddon, Anita Weatherby, and Dan Wright. Making Data a

First Class Scientific Output: Data Citation and Publication by NERC's Environmental Data Centres. *International Journal of Digital Curation*, 7(1):107–113, March 2012.

32. Leonardo Candela, Donatella Castelli, Paolo Manghi, and Alice Tani. Data journals: A survey. *Journal of the Association for Information Science and Technology*, n/a(n/a), 2015.
33. Leonardo Candela, Donatella Castelli, and Pasquale Pagano. D4science: an e-infrastructure for supporting virtual research environments. In *IRCDL*, pages 166–169. Citeseer, 2009.
34. Leonardo Candela, Paolo Manghi, Donatella Castelli, and Alice Tani. Data journals: A survey. *Journal of the Association for Information Science and Technology*, to appear, 2014.
35. Lucian Carata, Sherif Akoush, Nikilesh Balakrishnan, Thomas Bytheway, Ripduman Sohan, Margo Seltzer, and Andy Hopper. A primer on provenance. *Queue*, 12(3):10:10–10:23, March 2014.
36. Donatella Castelli, Paolo Manghi, and Costantino Thanos. A vision towards scientific communication infrastructures. *International Journal on Digital Libraries*, 13(3-4):155–169, 2013.
37. Chaomei Chen. Citespace ii: Detecting and visualizing emerging trends and transient patterns in scientific literature. *Journal of the American Society for Information Science and Technology*, 57(3):359–377, 2006.
38. CODATA-ICSTIT. Out of cite, out of mind: The current state of practice, policy, and technology for the citation of data. *Data Science Journal*, 12:CIDCR1–CIDCR75, 2013.
39. Rodrigo Costas, Ingeborg Meijer, Zohreh Zahedi, and Paul Wouters. The value of research - data metrics for datasets from a cultural and technical point of view. Knowledge exchange report, Knowledge Exchange, April 2013.
40. Mark J. Costello, William K. Michener, Mark Gahegan, Zhi-Qiang Zhang, and Philip E. Bourne. Biodiversity data should be published, cited, and peer reviewed. *Trends in Ecology & Evolution*, 28(8):454–461, 2013.
41. David De Roure, Carole Goble, and Robert Stevens. The design and realisation of the myExperiment Virtual Research Environment for social sharing of workflows. *Future Generation Computer Systems*, 25(5):561–567, May 2009.
42. Elaine Devine. Making data beautiful: the importance of supplemental material. *Digital Science Blog*, November 2014.

43. Anhai Doan, Raghu Ramakrishnan, and Alon Y. Halevy. Crowdsourcing systems on the World-Wide Web. *Commun. ACM*, 54(4):86–96, April 2011.

44. Kimberly Douglass, Suzie Allard, Carol Tenopir, Lei Wu, and Mike Frame. Managing scientific data as public assets: Data sharing practices and policies among full-time government employees. *Journal of the Association for Information Science and Technology*, 65(2):251–262, 2014.

45. Thomas Erl. *Soa: principles of service design*, volume 1. Prentice Hall Upper Saddle River, 2008.

46. Kristin R. Eschenfelder and Andrew Johnson. Managing the data commons: Controlled sharing of scholarly data. *Journal of the Association for Information Science and Technology*, 65(9):1757–1774, 2014.

47. Andrea Esuli. Automatic generation of lexical resources for opinion mining: Models, algorithms and applications. *SIGIR Forum*, 42(2):105–106, November 2008.

48. Benedikt Fecher and Sascha Friesike. Open science: One term, five schools of thought. In Sönke Bartling and Sascha Friesike, editors, *Opening Science*, pages 17–47. Springer International Publishing, 2014.

49. David F. Ferraiolo, D. Richard Kuhn, and Ramaswamy Chandramouli. *Role-Based Access Control*. Artech House, Inc., Norwood, MA, USA, 2003.

50. S. E. Fienberg, M. E. Martin, and M. L. Straf, editors. *Sharing research data*. The National Academies Press, 1985.

51. Force11. Data citation principles, retrieved from http://www.force11.org/datacitation, 11 2013.

52. Philippa Gardner, Nobuko Yoshida, Michele Bugliesi, Dario Colazzo, and Silvia Crafa. *Lecture Notes in Computer Science*, volume 3170, pages 225–239. Springer Berlin Heidelberg, 2004.

53. Jim Gray, David T. Liu, Maria N. Santisteban, Alex Szalay, David J. DeWitt, and Gerd Heber. Scientific data management in the coming decade. *SIGMOD Rec.*, 34(4):34–41, December 2005.

54. V. Henning and J. Reichelt. Mendeley - A Last.fm For Research? In *eScience, 2008. eScience &#039;08. IEEE Fourth International Conference on*, pages 327–328. IEEE, December 2008.

55. Tony Hey, Stewart Tansley, and Kristin Tolle. *The Fourth Paradigm: Data-Intensive Scientific Discovery*. Microsoft Research, 2009.

56. Michael Kircher and Prashant Jain. *Pattern-Oriented Software Architecture Volume 3: Patterns for Resource Management*. Wiley, June 2004.

57. Jens Klump, Roland Bertelmann, Jan Brase, Michael Diepenbroek, Hannes Grobe, Heinke Höck, Michael Lautenschlager, Uwe Schindler, Irina Sens, and Joachim Wächter. Data publication in the open access initiative. *Data Science Journal*, 5:79–83, 2006.

58. Stacy Kowalczyk and Kalpana Shankar. Data sharing in the sciences. *Annual Review of Information Science and Technology*, 45(1):247–294, 2011.

59. J. Kratz. Fifteen ideas about data validation (and peer review), http://datapub.cdlib.org/2014/05/08/fifteen-ideas-about-data-validation-and-peer-review.

60. J. Kratz and C. Strasser. Data publication consensus and controversies. *F1000Research,*, 94(3), 2014.

61. C. Lagoze, D. B. Krafft, S. Payette, and S. Jesuroga. What Is a Digital Library Anyway? Beyond Search and Access in the NSDL. *D-Lib Magazine*, 11(11), November 2005.

62. Bryan Lawrence, Catherine Jones, Brian Matthews, Sam Pepler, and Sarah Callaghan. Citation and Peer Review of Data: Moving Towards Formal Data Publication. *International Journal of Digital Curation*, 6(2), 2011.

63. Peter A. Lawrence. The politics of publication. authors, reviewers and editors must act to protect the quality of research. *Nature*, 422:259–261, 2003.

64. Carole J. Lee, Cassidy R. Sugimoto, Guo Zhang, and Blaise Cronin. Bias in peer review. *Journal of the Association for Information Science and Technology*, 64(1):2–17, 2013.

65. Hakan Lindqvist. Mandatory access control. 2006.

66. Paolo Manghi, Lukasz Bolikowski, Natalia Manola, Jochen Schirrwagen, and Tim Smith. OpenAIREplus: the european scholarly communication data infrastructure. *D-Lib Magazine*, 18(9/10), 2012.

67. Sara Mannheimer, Ayoung Yoon, Jane Greenberg, Elena Feinstein, and Ryan Scherle. A balancing act: The ideal and the realistic in developing dryad's preservation policy. *First Monday*, 19(8), 2014.

68. Laura Haak Marcial and Bradley M. Hemminger. Scientific data repositories on the web: An initial survey. *Journal of the American Society for Information Science and Technology*, 61(10):2029–2048, 2010.

69. Ethan Marcotte. *Responsive web design*. Editions Eyrolles, 2011.

70. Matthew S. Mayernik, Sarah Callaghan, Roland Leigh, Jonathan Tedds, and Steven Worley. Peer review of datasets: When, why, and how. *Bulletin of the American Meteorological Society*, 2015/01/12 2014.

71. Gary McGath. The format registry problem. *Code4Lib*, 19, 2013.

72. H Mooney and MP. Newton. The anatomy of a data citation: Discovery, reuse, and credit. *Journal of Librarianship and Scholarly Communication*, 1(1):eP1035, 2012.

73. Hailey Mooney and Mark P. Newton. The anatomy of a data citation: Discovery, reuse, and credit. *Journal of Librarianship and Scholarly Communication*, 1(1):eP1035, 2012.

74. Brian A. Nosek and Yoav Bar-Anan. Scientific communication is changing and scientists should lead the way. *Psychological Inquiry*, 23(3):308–314, 2012.

75. Brian A. Nosek and Yoav Bar-Anan. Scientific Utopia: I. Opening Scientific Communication. *Psychological Inquiry*, 23:217–243, 2012.

76. Joseph D. Novak and D. Bob Gowin. *Learning How to Learn*. Cambridge University Press, 1984.

77. OpenAIRE. https://www.openaire.eu/en/component/content/article/76-highlights/344-a-short-introduction-to-enhanced-publications.

78. Pasquale Pagano, Leonardo Candela, and Donatella Castelli. Data interoperability. *CODATA Data Science Journal*, 12:GRDI19–GRDI25, 2013.

79. Carole L. Palmer, Melissa H. Cragin, P. Bryan Heidorn, and Linda C. Smith. Data curation for the long tail of science: The case of environmental sciences. In *In Third International Digital Curation Conference, Washington, DC*, 2007.

80. Heinz Pampel and Sünje Dallmeier-Tiessen. Open research data: From vision to practice. In Sönke Bartling and Sascha Friesike, editors, *Opening Science*, pages 213–224. Springer International Publishing, 2014.

81. M A Parsons and P A Fox. Is data publication the right metaphor? *Data Science Journal*, 12, 2013.

82. Heather Piwowar. Altmetrics: Value all research products. *Nature*, 493(7431):159–159, 01 2013.

83. Heather A. Piwowar, Roger S. Day, and Douglas B. Fridsma. Sharing detailed research data is associated with increased citation rate. *PLoS ONE*, 2007.

84. Heather A. Piwowar and Todd J. Vision. Data reuse and the open data citation advantage. *PeerJ*, 1(e175), 2013.

85. Allen H. Renear, Simone Sacchi, and Karen M. Wickett. Definitions of dataset in the scientific and technical literature. *Proceedings of the American Society for Information Science and Technology*, 47(1):1–4, 2010.

86. EUROPEAN COMMISSION DIRECTORATES-GENERAL FOR RESEARCH, INNOVATION (RTD), CONTENT COMMUNICATIONS NETWORKS, and TECHNOLOGY (CONNECT). Consultation on ‘science 2.0’: Science in transition, 11 2014.

87. Dimitris Rousidis, Emmanouel Garoufallou, Panos Balatsoukas, and Miguel-Angel Sicilia. Data quality issues and content analysis for research data repositories : The case of dryad. In *ELPUB2014*, pages 45–98. IOS Press, 2014.

88. Douglas C. Schmidt, Hans Rohnert, Michael Stal, and Dieter Schultz. *Pattern-Oriented Software Architecture: Patterns for Concurrent and Networked Objects*. John Wiley &amp; Sons, Inc., New York, NY, USA, 2nd edition, 2000.

89. M. T. Schmidt, B. Hutchison, P. Lambros, and R. Phippen. The Enterprise Service Bus: Making service-oriented architecture real. *IBM Systems Journal*, 44(4):781–797, 2005.

90. Joan Starr and Angela Gastl. isCitedBy: A metadata scheme for DataCite. *D-Lib Magazine*, 17(1/2), January 2011.

91. SURFfoundation. https://www.surf.nl/en/themes/research/research-data-management/enhanced-publications/index.html.

92. Carol Tenopir, Suzie Allard, Kimberly Douglass, Arsev Umur Aydinoglu, Lei Wu, Eleanor Read, Maribeth Manoff, and Mike Frame. Data sharing by scientists: Practices and perceptions. *PLoS ONE*, 6(6):e21101, 2011.

93. Fei-Yue Wang, K.M. Carley, Daniel Zeng, and Wenji Mao. Social computing: From social informatics to social intelligence. *Intelligent Systems, IEEE*, 22(2):79–83, March 2007.

94. Richard Y. Wang and Diane M. Strong. Beyond accuracy: What data quality means to data consumers. *J. Manage. Inf. Syst.*, 12(4):5–33, March 1996.

95. Tim Wellhausen and Andreas Fiesser. How to write a pattern?: A rough guide for first-time pattern authors. In *Proceedings of the 16th European Conference on Pattern Languages of Programs*, EuroPLoP ’11, pages 5:1–5:9, New York, NY, USA, 2012. ACM.

96. Craig Willis, Jane Greenberg, and Hollie White. Analysis and synthesis of metadata goals for scientific data. *J. Am. Soc. Inf. Sci. Technol.*, 63(8):1505–1520, August 2012.

97. Craig Willis, Jane Greenberg, and Hollie White. Analysis and synthesis of metadata goals for scientific data. *Journal of the American Society for Information Science and Technology*, 63(8):1505–1520, 2012.

98. B. Plale Y.L. Simmhan and D. Gannon. A survey of data provenance in e-science. *SIGMOD Record*, September 2005.

Acknowledgments

First and foremost, I wish to thank my supervisor, Dr. Pasquale Pagano, who allowed me to undertake my Ph.D. at the Institute of Information Science and Technologies (ISTI) of the Italian National Research Council (CNR). Pasquale's constant encouragement, inspiration, and support led me to believe in the importance of this work.

I'm indebted to my supervisor Prof.ssa Cinzia Bernardeschi for the advices and the support she provided me during this work.

Thanks to the members of the Networked Multimedia Information Systems Laboratory of the ISTI - CNR, I benefited from useful discussions and friendly collaborations with each of them during the day-by-day work. Special thanks go to Leonardo Candela and Paolo Manghi for the advices and the support they provided me during this work. Many thanks go to Donatella Castelli, she was able to set up, during these years, a research group in which an exciting scientific activity is performed in a really special and friendly surrounding.

Special thanks go to Raffaele and Biagina, my parents, my "nonna" Antonietta, my sister Michela, and my brother-in-law Marco. They represent my family and supported me with love and patience during these years. This achievement belongs also to them. Last but not least, I thank Elisa, for her sustainment during the period of this work and for her capacity to fill my life with joy.