Realizing a Scalable and History-aware Literature Broker Service for OpenAIRE

Paolo Manghi, Claudio Atzori, Alessia Bardi,
Sandro La Bruzzo, Michele Artini

Consiglio Nazionale delle Ricerche
Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo"
Via Moruzzi 1, 56124 Pisa, Italy

name.surname@isti.cnr.it

Abstract. The OpenAIRE infrastructure is the point of reference for Open Science in Europe. Its services populate and provide access to a graph of objects relative to publications, datasets, people, organizations, projects, and funders aggregated from a variety of data sources, such as institutional repositories, data archives, journals, and CRIS systems. Not only, objects in the graph are harmonized to achieve semantic homogeneity, de-duplicated and merged, and enriched by inference with missing properties and/or relationships. The OpenAIRE Literature Broker Service is designed to offer subscription and notification functionalities for institutional repositories to: (i) learn about publication objects in OpenAIRE that do not appear in their collection but may be pertinent to it, and (ii) learn about extra properties or relationships relative to publication objects in their collection. Due to the high variability of the information space the following problems may arise: (i) subscriptions may vary over time to adapt to information space evolution, (ii) repository managers need to be able to quickly test their configurations before activating them, (iii) notifications may be redundant, and (iv) notifications may be very large over time. This paper presents the data model and software architecture of the OLBS, specifically designed to address these issues.

Keywords: subscription and notification, scalability, architecture, OpenAIRE, literature, publications, broker, services

1 Introduction

The OpenAIRE initiative [5] is the point of reference for Open Access in Europe. Its mission is to foster an Open Science e-Infrastructure that links people, ideas and resources for the free flow, access, sharing, and re-use of research outcomes, services and processes for the advancement of research and the dissemination of scientific

knowledge. OpenAIRE operates an open, participatory, service-oriented infrastructure that supports:

- The realization of a pan-European network for the definition, promotion and implementation of shared interoperability guidelines and best practices for managing, sharing, re-using, and preserving research outcomes of different typologies;
- The promotion of Open Science policies and practices at all stages of the research life-cycle and across research communities belonging to different application domains and geographical areas;
- The discovery of and access to research outcomes via a centralized entry point, where research outcomes are enriched with contextual information via links to objects relevant to the research life-cycle;
- The measurements of the impact of Open Science and the return of investment of national and international funding agencies.

Its technological infrastructure provides services [7] that populate the so-called OpenAIRE Information Space, a graph-like information space aggregating information about publications, datasets, organizations, persons, projects and several funders (e.g. European Commission, Wellcome Trust, Fundação para a Ciência e a Tecnologia, Australian Research Council) collected from hundreds of online data sources (e.g. publication repositories, dataset repositories, CRIS systems, journals, publishers).

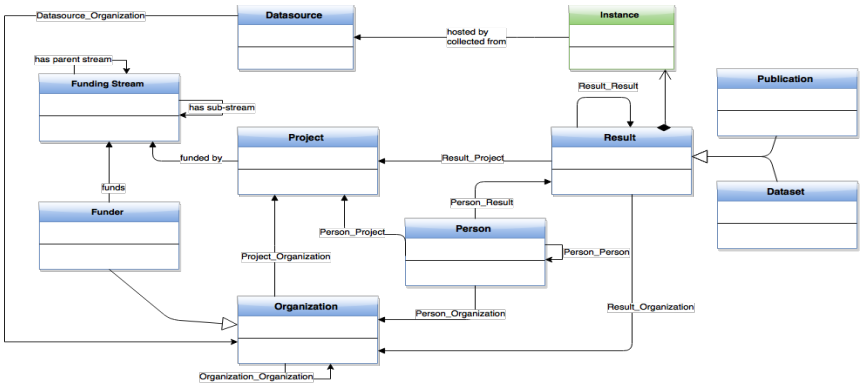

Figure 1 - The OpenAIRE data model

The OpenAIRE Information Space, whose data model [10] is shown in Figure 1, is obtained via the combined effort of three infrastructure sub-systems, depicted in Figure 2:

- *Harmonization* (aggregation sub-system): The OpenAIRE infrastructure collects metadata records from data sources and derives from them objects and relationships that form the information space graph (typologies and numbers of data sources currently included in OpenAIRE are available from https://www.openaire.eu/search/data-providers). For example a bibliographic

metadata record describing a scientific article will yield one publication object and a set of person objects (one per author) with relationships between them. Objects of given entities are transformed from their native data models (e.g. physically represented as XML records, HTML responses, CSV files) onto the OpenAIRE data model [10] in order to build an homogenous information space;

- *Merge* (de-duplication subsystem): objects of the same entity type are de-duplicated in order to remove ambiguities that may compromise statistics and impact (e.g. the same publication may be collected from different repositories as supposedly different objects)
- *Enrichment* (information inference sub-system): publication full-texts are collected and processed by text mining services [12] capable of inferring new property values or new relationships between objects.

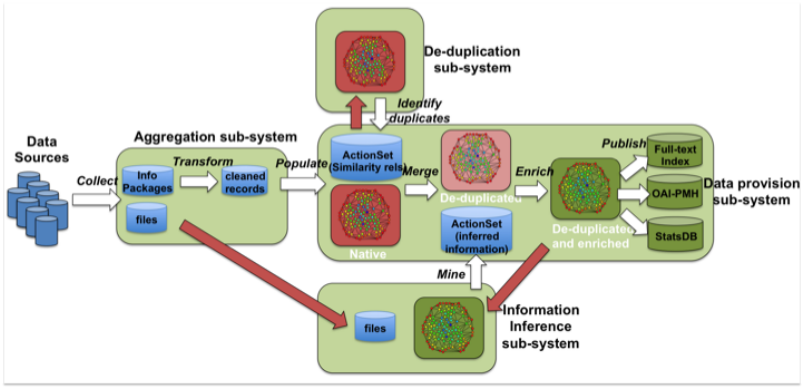

Figure 2 - OpenAIRE services high-level architecture

In order to give visibility to the contributing data sources, OpenAIRE keeps provenance information about each piece of aggregated information. Specifically, since de-duplication merges objects collected from different sources and inference enriches such objects, provenance information is kept at the granularity of the object itself, its properties, and its relationships. Object level provenance tells the origin of the object that is the data sources from which its different manifestations were collected. Property and relationship level provenance tells the origin of a specific property or relationship when inference algorithms derive these, e.g. algorithm name and version. The OpenAIRE Information Space is then made available for programmatic access via several APIs (Search HTTP APIs, OAI-PMH, and soon Linked Open Data) [2] and for search, browse and statistics consultation via the OpenAIRE portal (www.openaire.eu).

Data sources providing content to OpenAIRE and interested in augmenting their local collections may benefit in a number of ways from the OpenAIRE information space. This is particularly true for institutional repositories, whose mission is that of

growing a complete collection of the scientific publications produced by the authors affiliated to the institution they serve. Repository managers' goal is twofold: bringing in the collection all articles produced by such authors and making sure the metadata is as complete and up-to-date as possible. To this aim, the infrastructure is currently being equipped with the OpenAIRE Literature Broker Service (OLBS) whose general principles and ideas are described in [16]. The OLBS implements a subscription and notification mechanism supporting repository managers at enhancing the content of their repository taking advantage of the OpenAIRE information space. As reported in [16], a number of initiatives started working on "brokering" approaches favoring single-deposition of publication metadata with subsequent automated delivery to other repositories. Some focused on techniques for automatic deposition into a repository (SWORD project [4]), while others focused on the complementary aspects of how to broker publication information from publishers to relevant/interested repositories. SHARE [3] and JISC/EDINA [9] are two of such initiatives, based respectively in the U.S. and U.K. The OpenAIRE Literature Broker Service (OLB Service) offers to repository managers the possibility to subscribe to special "addition" or "enrichment" events, in order to be respectively notified about: *(i)* publication objects in OpenAIRE that do not appear in their collection but may be pertinent to it, or *(ii)* properties or relationships relative to publication objects in their collection that do not appear in their local metadata. Due to the high variability of the OpenAIRE information space, where new data sources are continuously added or removed, harmonization rules, mining, and deduplication algorithms are refined, following problems may arise: *(i)* subscriptions may vary over time to adapt to information space evolution, *(ii)* repository managers need to be able to quickly test their configurations before activating them, *(iii)* the same notifications may be sent more than once, and *(iv)* notifications may be very large in number over time. This paper presents the OLBS data model and software architecture, specifically designed to address these issues.

2 OLBS functional requirements

The OLBS operates on top of the OpenAIRE information graph and supports repository managers with a Web Dashboard from which they can subscribe to (potential) "enrichment" and (potential) "addition" events occurring to the graph and of interest to their repository. Figure 3 shows how the OLBS integrates with the existing OpenAIRE infrastructure. Data sources are aggregated, de-duplicated and enriched by mining techniques so as to populate the OpenAIRE Information space graph. Whenever a new information space is generated, the OLBS explores the graph to detect if any of the active subscriptions finds a match and in such case notifications are generated, delivered, and archived.

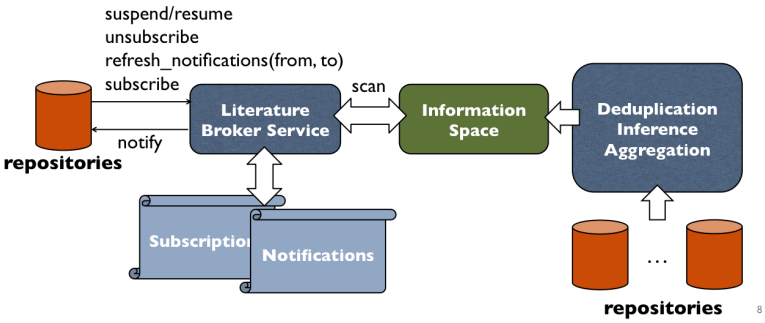

Figure 3 - The OLBS in the OpenAIRE infrastructure

2.1 Subscriptions

Repository managers will be able to subscribe to two main classes of subscriptions: "enrichment" and "addition".

Enrichment The first class refers to notifications about publications that (i) were collected from the repository by OpenAIRE and (ii) have been enriched with properties or relationships to other objects by OpenAIRE inference algorithms (e.g. relationships to projects and datasets, citation lists, document classification properties) or by the side effect of being merged with richer publication objects (e.g. DOI of a publication, Open Access version of the publication). The identification of these events is straightforward as it is based on provenance of collection (i.e. selects publication objects collected from the given repository) and of enrichment (i.e. further selects objects of the given repository involved into a merge or enriched by inference algorithms). Repository managers will be able to fine-tune their subscriptions based on the bibliographic fields they would like to be notified about; e.g. "return the fields DOIs and Funding Project relative to my records".

Addition The second class refers to notifications relative to publications that are "relevant to" the repository at hand, but are not present in the repository. The identification of these events requires the navigation of the Information Space graph, in an attempt to identify relationships between the subscribing institutional repositories (which are a specific OpenAIRE data source type) and publications that have not been collected from the given repository but are "relevant to" it. Three relationships have been identified, according to which a publication is relevant to a repository if one of the following chains of relationships exist in the OpenAIRE information graph:

- *Affiliation repository: publication-author-organization-repository*: the publication has an author whose organization (affiliation) has a given institutional repository of reference; the affiliation relationship *publication-author-*

organization is extracted by inference and has a level of trust *T:[0..1],* which represents the level of confidence of the inference algorithm for that specific statement; the relationship *organization-repository* is instead provided by an authoritative OpenAIRE data source, namely OpenDOAR[1], which maintains the directory of repositories and related responsible organizations;

- *Reference repository: publication-author-repository*: the publication has an author with a given institutional repository of reference; the relationship *author-repository* is inferred by mining the graph and identifying the correlations between authors, their co-authored publications and the repositories from which these publications were collected; each author can be associated to a repository with a weight of "repository reference-ness", obtained as the percentage of author publications occurring in the repository ("deposition rate"), and the repository with the highest percentage corresponds to the repository of reference for the author; the information space contains a relationship between authors and their inferred repository of reference, with a degree of trust *T:[0..1]* that depends on the "dominance" of the repository over other repositories (variables are total of author publications and total of related repositories).
- *Funder repository: publication-project-organization-repository*: the publication has been funded by a project whose participants (organizations beneficiaries of the grant) have a given institutional repository of reference; the relationship *publication-project* is either collected from the data sources or inferred by mining and has therefore a level of trust *T:[0..1]*; the relationship *organization-repository* is instead provided, as in the first case, by the OpenDOAR data source; with a degree of approximation, the repositories reached by such relationship may be interested in the given publication.

Given a publication, if one of such relationships exists in the graph (collected or inferred) the OLBS may notify the subscribing repositories of the publication. Since the relationships are generally not authoritative (not collected from data sources), but inferred by OpenAIRE services, subscribing repository managers can also fine-tune the minimal threshold of trust T_m for their subscriptions.

2.2 Notifications

If a repository manager activates several subscriptions or modifies over time the parameters of existing subscriptions, the same record may meet the criteria of different subscriptions and be notified several times to the repository. Repositories should therefore not be notified more than once of each relevant publication, unless explicitly requested. As a consequence, the OLBS should keep the history of all past notifications to allow repository managers to consult them and avoid their re-sending

[1] *The Directory of Open Access Repositories*, http://www.opendoar.org/

to the repositories. As described in [16], two different notification strategies are under evaluation in order to meet diverse requirements of subscribers:

- **Mail postcards** Subscribers may opt to be notified by email at given interval of times (e.g. daily, weekly, monthly) and with given granularity (individual records, digests, URL to a web user interface).
- **Programmatic access** *Pull mode*: APIs will be provided to retrieve notifications by status (e.g. read/unread), subscription typology, and filters (e.g. criteria on the metadata fields). *Push mode:* the SWORD [4] protocol for automatic ingestion of records into repositories will be evaluated.

2.3 Web Dashboard

Repository managers are supported with the OLBS Web Dashboard, from where they can activate and configure subscriptions of the available types, how publications of these categories should be notified, and also explore the history of notifications they have so far received (e.g. searching and browsing notifications by type, publication title, authors publication date, notification date). Repository managers can fine-tune their subscriptions by real-time interacting with the graph and test the quality of the notifications they would collect before they actually submit the subscription.

3 Data model and architecture

This section presents the data model specification, the high level architecture of the OLBS, and the proposed implementation aiming at satisfying the functional requirements described in section 2.

3.1 Data model

The data model is illustrated in Figure 4 and includes the classes: *repository, subscription, potential notification,* and *notification.* The model relies on a "topic tree"[2] that encodes the typologies of subscriptions supported by the OLBS, which are of two main classes *addition* and *enrichment*:

- *addition.*<*subclass*>: the values of <*subclass*> encode the three approaches for the identification of publications "relevant to" a repository, namely: *byAffiliation*, *byReference*, and *byFunder*.
- *enrichment.*<*subclass*>: the values of <*subclass*> encode the six publication enrichment notification use-cases: *open_access_version*, *project_link*, *dataset_link*, *subject classifications*, *DOI*, *author_PID*.

[2] *OASIS Standard WS-Topics specification* 1.3 (2006), http://docs.oasis-open.org/wsn/wsn-ws_topics-1.3-spec-os.pdf

Repository Institutional repository registered to the OLBS.
Subscription Subscriptions are associated to a *repository* and are characterized by the following configuration parameters:

- *trustThreshold*: the minimum value of trust in the interval [0..1] that a *potential notification* must satisfy to be included in the *notifications* of this subscription;
- *criteria: CQL* query identifying a further criteria to be respected by the set of publication fields included in *potential notification*;
- *type*: path in subscription tree identifying the *potential notifications* of interest;
- *notification_scheduling:* how often the notifications for this subscription must be generated and sent (daily, weekly, monthly);
- *notification_granularity*: what must be included in each notification (full Dublin Core of all publications, digest, or a URL to the web dashboard)
- *notification_mode:* possible options will depend on the OLBS functionalities, examples are email and SWORD protocol.

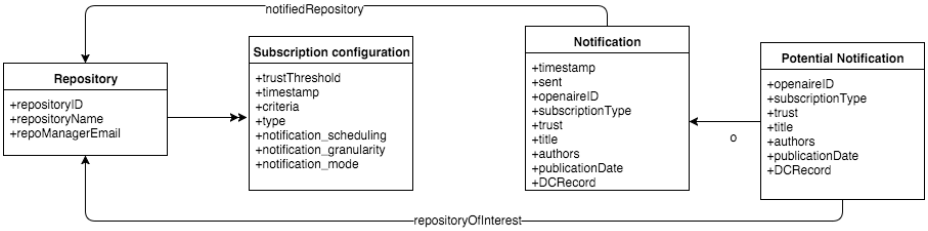

Figure 4 - Data model

Potential Notification Potential Notifications are generated by analyzing the OpenAIRE graph information space to identify the whole range of repository-publication pairs that may be of interest to repositories in OpenAIRE, i.e. independently from the existence of an active *subscription*. Each potential notification is associated to a *repository* and is characterized by the following properties:

- *openaireId*: the unique identifier of the publication in the information space;
- *title*, *authors*, *publicationDate* of the publication, to enable search and browse;
- *DCRecord*: the Dublin Core record of the publication;
- *subscriptionType*: path in the topic tree;
- *trust*: level of trust of publication-*repository* association w.r.t. subscription type;

Notification When a *potential notification* matches a *subscription*, a corresponding notification is created and the *repository* is alerted, respecting *scheduling* and *mode* of the relative *notification configuration*. A notification must be persisted as the evidence that a publication has been notified to a repository. As such, it is a copy of the relative *potential notification*, which, due to the variability of the information space, may not necessarily persist as long as the notifications it has generated

(publications in OpenAIRE may disappear). A notification includes all properties of the corresponding potential notification plus the *timeOfCreation* of the notification.

3.2 Architecture overview

The OpenAIRE information space is stored in an HBASE cluster (8 worker nodes, each of them with 8 CPUs and 24GB RAM) in order to support performance and scalability. Metadata records collected from data sources are harmonised and transformed into objects compliant to the OpenAIRE data model. The objects are then stored into HBASE, where each object is converted into one HBASE row. In December 2015, OpenAIRE collected more than 15,7M publication metadata records, corresponding to more than 30 millions OpenAIRE objects (and therefore HBASE rows). Once populated, the HBASE table is ready to be processed by inference and de-duplication algorithms for enrichment:

- The inference subsystem analyzes the OpenAIRE information space, supported by the available full-texts of publications to generate properties and relationships, including the *relationships publication-XXX-repository* needed by the three subscription methods mentioned in Section 2.1. In December 2015, the inference subsystem enriched more than 1,2M publications (8% of the total publications) with relationships and properties.
- The de-duplication subsystem runs Mapreduce jobs on the HBASE table that implement heuristics for the detection of duplicates of publications, organisations and persons [17]. Groups of duplicate objects are then merged into one disambiguated object. Depending on the configuration settings, the deduplication process for 15M publications takes 2-3 hours, identifying 4,1M of duplicates and merging them into 1,7M of disambiguated objects.

The enriched HBASE table is then processed for publishing via the OpenAIRE portal and standard APIs [2]. The OLBS will have to further process the OpenAIRE information space stored on HBASE to identify addition and enrichment events to be notified to subscribers. Figure 5 illustrates the OLBS integration in the OpenAIRE infrastructure, which consists of a three-phase data workflow.

Phase 1: generation of potential notifications Whenever the OpenAIRE infrastructure generates a new version of the information space (i.e. hamornisation, deduplication, enrichment steps are performed), the OLBS will run MapReduce jobs on the HBASE table to identify the current *potential notifications.* These jobs produce tuples of the form: `openaireID`, `SubscriptionType`, `repositoryId`, `trust`, `title`, `authors`, `PublicationDate`, `DCRecord`. For each publication-repository pair such that: *(i)* the publication was collected from the repository but is richer in properties or relationships in OpenAIRE (*trust* corresponds to the level of *trust* of the OpenAIRE enrichment), or *(ii)* there exist a chains of relationships *addition.affiliation,*

addition.reference, or *addition.funder* between the two (*trust* corresponds to a combination of the inferred relationship *trust* level and other parameters, see section 2.1; the formulae will be refined over time).

Scalability challenges Potential notifications are at the core of the system, since they need to be often updated and searched for matching with subscriptions, and may reach very large numbers, which can be estimated as follows:

- Additions: institutional repositories aim at collecting all publications of authors of reference, hence the assumption that repositories miss an average of 20% of their publications makes a reasonable worst-case scenario. Since OpenAIRE counts around 400 institutional repositories with an average of 13,000 publications, the estimate of missing publications is around 1M in the worst case.
- Enrichments: the number of institutional repository publications subject to de-duplication and therefore possibly affected by enrichment is around 1,4M; inference is applied to publications with a PDF, which are today around 3M. In summary, the worst-case envisages 4,4M multiplied by the 6 potential "enrichment" notifications, for a total of around 27M.

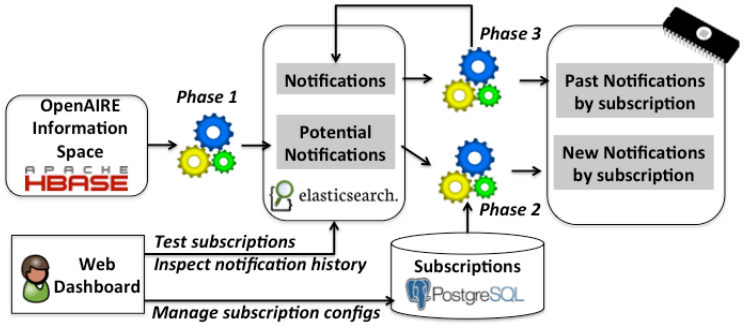

Figure 5 - OLB Data Flow

Hence, in the optimistic assumption that OpenAIRE would aggregate and elaborate all missing publications and all possible enrichments, the order of magnitude of potential notifications is overall around 28M. In order to make them searchable (phase 2 and 3 below) the OLBS must build on a scalable back-end, efficient on dropping and feeding entries, and also capable of supporting efficient queries. The current implementation plan finds in Elasticsearch[3] full-text index the best fitting candidate among the Open Source search engines due to its capability of scaling up horizontally and the expressivity of its data model and query language.

Phase 2: subscription matching

[3] Elasticsearch *https://www.elastic.co*

Subscriptions, managed by repository managers via the Web Dashboard, are stored on a relational database (PostgreSql[4]). Based on the *notification_schedule* property of the available subscriptions, the OLBS searches potential notifications matching the criteria of the existing subscriptions: repository, subscription type, criteria, and minimal level of trust. The resulting potential notifications are temporarily kept in memory, ready to be used in the following phase.

Phase 3: find new notifications

The Elasticsearch index contains another collection of *notifications*, which tracks all *potential notifications* that matched a subscription and have been sent to the relative repositories. Notifications are entries of the form: `timestamp`, `openaireID`, `Subscription Type`, `repositoryId`, `trust`, `title`, `Authors`, `Publication Date`, `DCRecord`; where timestamp is the notification date (the same for all notifications identified and sent in the same Phase 2 session). Starting from the potential notifications identified in Phase 2, currently kept in memory, in Phase 3 the OLBS searches the *notifications* collection in order select which *potential notifications* are eligible to become new *notifications*, thereby avoiding multiple delivery of the same notifications to the same repositories. Only new notifications will be fed to the index and be sent to the repositories according to the requested notification strategy.

4 Conclusions

The first BETA instantiation of the OLBS will be available by June 2016. The next steps are the generalization of the core of the OLBS to become a general-purpose Literature Broker Service, ready to support the same functionalities independently of the context. To this aim, the data model and internal representations of the OLBS, together with the acquired experience in operating the service, will inspire the definition of standard exchange formats and APIs to achieve interoperability across a network of similar services world-wide (collaborations are active with JISC-UK and SHARE-US). Concepts such as notification broker services, literature subscriptions/notifications descriptions, subscription/notification APIs should be agreed on, conform to existing standards[5], and shared across the community and become a common way to share objects across repositories.

Acknowledgement This work is partially funded by the European Commission H2020 project OpenAIRE2020 (Grant agreement: 643410, Call: H2020-EINFRA-2014-1).

[4] PostgreSql *http://www.postgresql.org*

[5] OASIS Web Services Notification (WSN), https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsn (26/01/2016)

5 References

1. The OpenAIRE guidelines. Available at: https://guidelines.openaire.eu. Last accessed: 10 July 2015.
2. The OpenAIRE API. Available at: https://api.openaire.eu. Last accessed: 10 July 2015.
3. Walters, T., & Ruttenberg, J. (2014). SHared Access Research Ecosystem. Educause Review, 49(2), 56-57. Available at: http://www.educause.edu/ero/article/shared-access-research-ecosystem. Last accessed: 10 July 2015.
4. Lewis, S., de Castro, P., & Jones, R. (2012). SWORD: Facilitating deposit scenarios. D-Lib Magazine, 18(1), 4. doi:10.1045/january2012-lewis
5. Manghi, P., Bolikowski, L., Manold, N., Schirrwagen, J., & Smith, T. (2012). Openaireplus: the european scholarly communication data infrastructure. D-Lib Magazine, 18(9), 1. doi:10.1045/september2012-manghi
6. What is the Open Research Data Pilot?. Available at: https://www.openaire.eu/ordp/ordp/pilot. Last accessed: 12 July 2015
7. Manghi, P., Artini, M., Atzori, C., Bardi, A., Mannocci, A., La Bruzzo, S., Candela L., Castelli D., & Pagano P. (2014). The D-NET software toolkit: A framework for the realization, maintenance, and operation of aggregative infrastructures. Program: electronic library and information systems, 48(4), 322-354. doi:10.1108/PROG-08-2013-0045
8. COAR: Confederation of Open Access Repositories. Available at: https://www.coar-repositories.org/. Last accessed: 12 July 2015.
9. The JISC website. Available at: https://jisc.ac.uk/. Last accessed: 12 July 2015.
10. Manghi, P., Houssos, N., Mikulicic, M., & Jörg, B. (2012). The data model of the openaire scientific communication e-infrastructure. In Metadata and Semantics Research (pp. 168-180). Springer Berlin Heidelberg. doi:10.1007/978-3-642-35233-1_18
11. Houssos, N., Jörg, B., Dvořák, J., Príncipe, P., Rodrigues, E., Manghi, P., & Elbæk, M. K. (2014). OpenAIRE guidelines for CRIS managers: supporting interoperability of open research information through established standards. Procedia Computer Science, 33, 33-38. doi:10.1016/j.procs.2014.06.006
12. Kobos, M., Bolikowski, Ł., Horst, M., Manghi, P., Manola, N., & Schirrwagen, J. (2014). Information Inference in Scholarly Communication Infrastructures: The OpenAIREplus Project Experience. Procedia Computer Science, 38, 92-99. doi:10.1016/j.procs.2014.10.016
13. DataCite web site. Available at: https://www.datacite.org. Last accessed: 12 July 2015.
14. The CERIF data model. Available at: http://www.eurocris.org/cerif/main-features-cerif. Last accessed: 10 July 2015.
15. Jisc Blog, "Jisc Publications Router enters a new phase", http://scholarlycommunications.jiscinvolve.org/wp/2015/07/01/jisc-publications-router-enters-a-new-phase. Last accessed 21 July 2015.
16. Artini, M., Atzori, C., Bardi, A., La Bruzzo, S., Manghi, P., & Mannocci, A. (2015). The OpenAIRE Literature Broker Service for Institutional Repositories. D-Lib Magazine, 21(11), 3.
17. Manghi, Paolo, and Marko Mikulicic. "PACE: A general-purpose tool for authority control." *Metadata and Semantic Research*. Springer Berlin Heidelberg, 2011. 80-92