Development of a Federated Repository Infrastructure for the Arts and Humanities in Germany (R 1.3.1)

Version – 1.0

Work Package – 1

Responsible Partner – SUB Göttingen

TextGrid

Virtual Research Environment for the Humanities

eScience methods in Arts and Humanities

Projekt: TextGrid – Virtual Research Environment for the Humanities

Funded by the German Federal Ministry of Education and Research (BMBF) by Agreement: 01UG0901A

Project Duration: June 2009 – May 2012

Dokument Status: final

Distribution: public

Authors:

Andreas Aschenbrenner (SUB)

Wolfgang Pempe (SUB)

Table of Contents

1. Motivation

TextGrid is not only a Virtual Research Environment but also an infrastructure enabling the collective utilisation and exchange of data, tools and methods. One of the most important components of this infrastructure is a Grid-based repository ensuring the sustainable availability of and access to research data, the so-called *TextGrid Rep*. While the tool development and the user interface (*TextGrid Lab*) naturally has to focus on the needs of only a few disciplines (German literature and linguistics, classical philology, musicology, history of arts), the TextGrid infrastructure is designed to be as open and flexible as possible – in order to play a part in the larger world of digital ecosystems [5, 43] – "open, loosely coupled, demand-driven, domain clustered, agent-based self organized collaborative environment[s] where species/agents form a temporary coalition (or longer term) for a specific purpose or goals, and everyone is proactive and responsive for its own benefit or profit" [23].

If anything characterises Arts and Humanities' research best, then it is diversity and heterogeneity – contrasting (or better: amending) the frequently invoked *data deluge* [1, 37] with a *complexity deluge* [15, 27]. Federating repositories – and more or less directly – attached VREs, fosters heterogeneity, while enabling interoperability between diverse repository systems and other agents in an open repository environment[1].

The two principles – open and generic (TextGrid Rep) vs. specialised (TextGrid Lab) – may seem to be contradictory at first sight, and they reflect the contradictions in the original requirements for enabling idiosyncratic research questions and methodologies while fostering collaboration and interoperability. However, they are only contradictory at first sight, when aiming to build a comprehensive system that fulfils all the posed requirements. Yet, TextGrid is not so much a system as rather an open platform that enables scholars to adapt the environment to their needs.

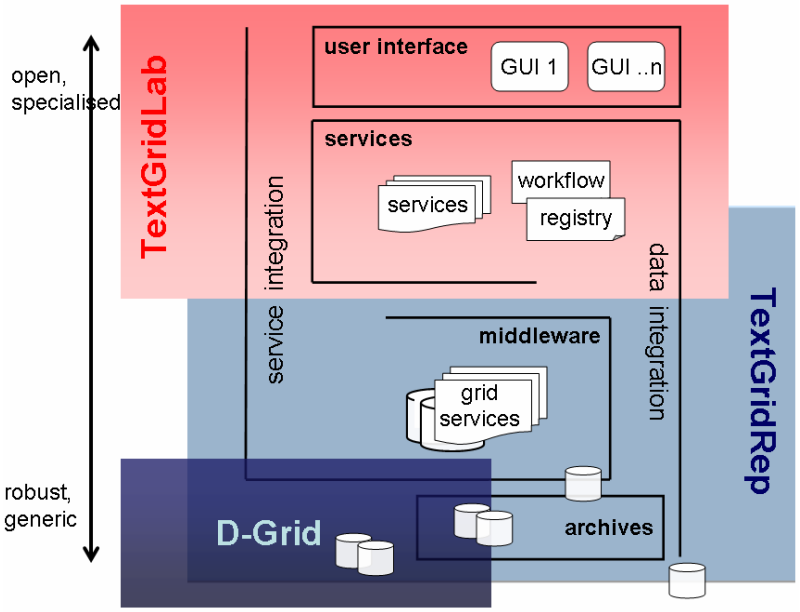


Figure 1. TextGrid – Architecture

Tharam Dillon et al. identify the following three features as general architectural design principles for open environments: loosely-coupled, simple, and decentralised (cf. the "Evolution-

[1] Most of the following text passages are excerpts from [3].

ary CUBE", [26]). Frank Buschmann et al. support this with two similar attributes for quality interfaces (with respect to Interface Partitioning): "expressiveness and simplicity", as well as "loose coupling and stability". [18]

> **loosely-coupled** – "The core principle behind loose coupling is to reduce the assumptions two parties (components, applications, services, programs, users) make about each other when they exchange information." (cf. [41], page 10) Reduction of assumptions comes in many interrelated facets ([49] identified 12 such facets). When optimising along these facets and thus raising the degree of loose-coupling, systems become more extensible and have the potential to grow and scale rapidly – characteristics displayed e.g. by the RESTful architectural style. [31]
>
> **simple** – Simplicity manifests in a focused set of capabilities and stripped-down interfaces. This may be achieved e.g. by pruning complexity, by taking assumptions between the two parties (which works against the previous point on loose-coupling), or by decomposing complexity into simple modules moving complexity from a single service to the overall system.
>
> **decentralised** – Both, loose-coupling and simplicity further the independence of individual components, avoid lock-in into a specific component and enable the components to evolve independent from each other. This applies for interaction between specific components in a designed system, and it equally applies for external components. It is the link between internal and external components that is changing as repositories embed external infrastructure and added-value services. Vice versa, an open (i.e. loosely-coupled, simple, and decentralised) design allows repository-based applications and other components to interact with an existing system, thus enabling its anarchic growth. Following the mantra of the Common Repositories Interface Group (CRIG) [42]: "The coolest thing to do with your data [and services] will be thought of by someone else."

These three values are the basis for moving from a single integrated repository system to a larger, open repository environment, since they facilitate the interaction of multiple, decentralised agents (repositories, added-value services, repository-based applications, etc.).

Based on these considerations, the repository reference architecture (cf. figure 2) consists of three layers: virtualized storage at the bottom, upon which a layer for digital object management mediates to end-user applications. Note that none of the layers is called "repository", since the repository really is distributed across the layers. Each layer adds another level of abstraction to the content named as physical, logical, and conceptual, which is inspired by Thibodeau [66] and is reflected in the federation interface as well (see below).

The interfaces between the layers are more than merely conceptual borders of an architectural concept. It is these interfaces that enable mixing various repository components and external services in a decentralized manner to form a single repository environment. This means that an infrastructure for repository storage could serve multiple object management layers or vice versa, and equally any external service or end-user oriented application could build on one or many infrastructures and object management layers. This is essentially the kinds of scenarios presented below.

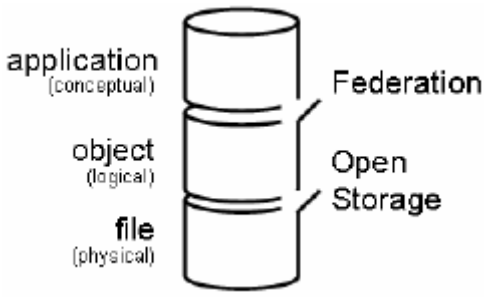


Figure 2. Schematic Repository Reference Architecture consisting of 3 layers (file, object, and application in rising abstraction), as well as two interfaces between the layers – the Open Storage and the Federation interfaces, which are the key interoperability channels of Open Repository Environments.

Federation mechanisms lie in between the object and the application layers. The federation interface actually is a cluster of mechanisms to achieve interoperability between diverse agents in an open repository environment. These mechanisms are capable of interweaving multiple repositories, respectively of enabling interaction between repositories and other agents.

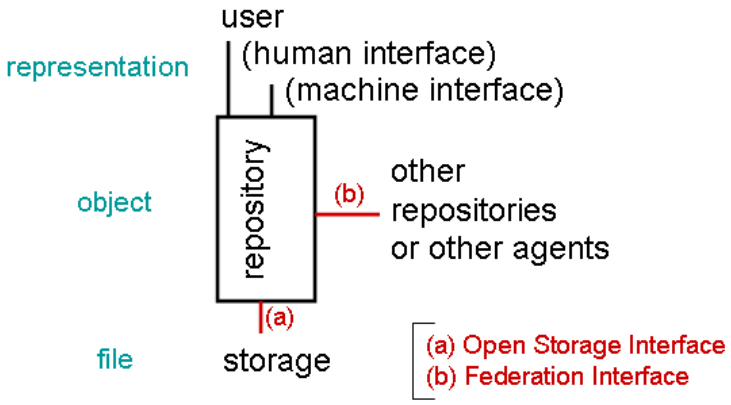


Figure 3. Open Storage and Federation Interfaces.

2. Federation Scenarios

Digital objects are often of interest in multiple contexts: publications may be disseminated through institutional as well as thematic repositories; research data may be created in a specific project and later re-used in another, maybe inter-disciplinary or inter-institutional project; and many other such situations are conceivable. A more detailed discussion in the next section analyses federation and develops novel federation mechanisms. In this section, we list three federation activities to give an idea of the kinds of environments enabled by federation.

The most prevalent use case for federation as yet is search across multiple repositories. Today many universities have their own institutional repositories. Initiatives like DARE[2] and DRIVER[3] establish central portals to search for publications on a national respectively European level. Other than the federations of research publications in DARE and DRIVER, the

[2] http://www.kb.nl/hrd/dd/dd_projecten/projecten_dare-en.html

[3] http://www.driver-repository.eu

Europeana[4] initiative addresses research data and aims to pool all digitisations of cultural material in Europe.

SDMX, the Statistical Data and Metadata eXchange[5], is "an initiative to foster standards for the exchange of statistical information", sponsored amongst others by national statistical offices, the World Bank, and the United Nations. Amongst the challenges for SDMX is the requirement to accommodate partners in remote areas (e.g. Africa), and to ensure that any updates even in remote countries are immediately propagated throughout the whole federation of global partners. To enable interoperability, SDMX includes metadata schemas as well as guidelines for web services [59] to interconnect statistical databases around the world.

The project TIPR, Towards Interoperable Preservation Repositories, federates preservation repositories including three university repositories based on heterogenous software platforms. As part of this federation, digital objects are replicated and distributed to the dispersed repositories. TIPR's goal is to ensure the longevity of the digital object. [20]

Other than in the case of the Open Storage Interface, there are various attempts for federating repositories on an object level. However, as the scenarios described below underline, current approaches are fragmented and often insufficient for contexts other than open access publication repositories. While there will never be a single, final solution to repository federation, we discuss an extended federation model in the next section. At this point we would only like to mention the two orthogonal types of federation and two prototypical and popular federation protocols:

Federated content – Combining multiple repositories in a single application increases both the exposure of the objects as well as the value of the application. Therefore, federation protocols have been created independently in various communities, including the following. Apart from protocols, metadata sets like Dublin Core[6], encapsulation formats like METS[7], schemas like PREMIS[8] or other standards are of relevance when federating repositories.

- Z39.50[9] for querying library catalogues has been developed in 1988, and became a NISO standard in 1992. The protocol was widely spread and still is. Its successor SRU/W[10] better suits the current web environment, and it is embedded in ongoing work for extending search/retrieve interfaces.
- The Protocol for Metadata Harvesting, OAI-PMH[11], was first released in 2001 to connect disparate library catalogues. Spurred by the open access movement [13] it quickly became a de facto standard. In September 2009, OAIster, a "union catalog" for digital resources[12], cross-referenced more than 1100 repositories by way of the OAI-PMH protocol and their more than 23 million digital resources. Apart from harvesting publications, OAI-PMH has been employed in other contexts as well ([24, 46, 56]).

---

[4] http://group.europeana.eu

[5] http://sdmx.org/

[6] http://dublincore.org

[7] http://www.loc.gov/standards/mets/

[8] http://www.loc.gov/standards/premis/

[9] http://www.cni.org/pub/NISO/docs/Z39.50-1992/

[10] http://www.loc.gov/standards/sru/

[11] http://www.openarchives.org/pmh/

[12] http://www.oclc.org/oaister/

Multiple applications – Embedding objects in multiple applications environments – the orthogonal federation mechanism to embedding objects from multiple repositories in a single application – has not found as much attention as its counterpart. Many repositories today offer interfaces or programming libraries to build custom applications on top of the repository infrastructure. However, there are as yet no standards that would enable an application to move from one repository platform to another. It may be argued to which extent that is useful and there will likely always be custom interfaces, yet some aspects may be covered by standards to ensure portability where needed.
The newly issued OAI-ORE standard[13] covers one aspect of this: object representation. OAI-ORE is a format specification for serialising digital objects expressed in RDF-based Resource Maps. Version 1.0 of OAI-ORE has been released in October 2008. Being the cousin of OAI-PMH, OAI-ORE has much attention guaranteed. Some of the future use cases it mentions include "applications that support authoring, deposit, exchange, visualization, reuse, and preservation."

The scenarios describe aspects of open repository environments, which build upon interweaving distinct repositories or outsourcing functionalities to external services and infrastructure. Existing repository federations (e.g. DRIVER, DARE, Europeana – see above) fail to satisfy the requirements put forth by open repository environments, since – as opposed to those traditional federation mechanisms – an open repository environment **(a)** deals with material that changes frequently and needs to propagate those changes in a timely manner, it **(b)** includes non-repository agents (e.g. format registries, migration services, visualisation of content networks), and **(c)** it enables interoperability on multiple layers of abstraction. [9] The following scenarios display all these features, and they are clustered along two interoperability levels, object storage and federation.

"In the future there will be only one (virtual) repository" – this is one of the visions for repository infrastructure formulated at the repository workshop at the Open Grid Forum Barcelona [7]. In fact, there are today various initiatives striving to federate physically distinct repositories into a single virtual repository, including DRIVER, DARE and Europeana. Content in those cases is dispersed over various locations for historical or for organizational reasons (e.g. each university library establishes its own institutional repository). In their integration efforts, the goal of all these initiatives is to build a single portal that provides access to these dispersed locations.

However, these initiatives predominantly focus on exchanging metadata about publications. In the case of the three initiatives mentioned above, all of them employ the prevalent Protocol for Metadata Harvesting of the Open Archives Initiative, OAI-PMH.The limitations of these kinds of federations are becoming apparent as repositories are increasingly managing research data (as opposed to publications), and multiple repositories are exchanging that research data for reuse (as opposed to only exchanging the metadata for viewing). [2, 10, 47] Also, the aforementioned federations usually just take whatever they can get. More fine-grained control over the federation may be required for building thematic collections composed of selected pieces from various repositories, or in the case of inter-disciplinary and inter-institutional projects.

In other words, the requirements on federation in open repository environments are very unlike traditional federation mechanisms. Various initiatives recognised this, including [17], who call for repository interoperability and Next Generation Services, which enable "deep sharing through experimentation with aggregation other than metadata harvesting, resulting in

[13] http://www.openarchives.org/ore/

the capacity to move digital objects from domain to domain, along with the ability to modify and re-deposit them in a different location in the process."

The following three scenarios discuss respectively the federation of data (scenario A), sharing metadata of frequently changing objects or collections (scenario B), as well as exchanging data with external, non-repository agents (scenario C).

Scenario A: Scientific Analysis

Particularly in the humanities, research is not confined to a single location but often includes material from dispersed locations. [8] Each of these locations may have an institutional repository with relevant material for a specific research question that bridges all those locations. In this scenario, these distinct repositories federate without changing the underlying technologies, offering search and analysis across their collections in a dedicated portal. With the emergence of more and more repository-based research environments, the need for scientific analysis of repository contents is likely to increase. [2, 46, 68] The kinds of analysis conducted by such a joint portal can be manifold. Federation mechanisms should not constrain analysis technologies, and they should not constrain the kind of objects to be shared both with regards to their content and their metadata.

In particular, we would like to point out two challenges that analysis functions may pose on the scalability of the overall system. First, an analysis technology could be very resource-intense even when applied to only a single repository, yet should not bring down the performance of the repository. Retrieval or clustering techniques are just two of the fields offering dedicated analysis methods that are very resource-intense, yet may be of interest to repository-based research environments. [53, 65] The second challenge mentioned here is that fast-changing content should not bring down the scalability of the overall system, even as many repositories join the federation. Fast-changing content requires an immediate link between the numerous repositories and their joint analysis portal to avoid inconsistencies, and may hence increase the communication demand significantly compared to immutable content.

Scenario B: Task Tracking

An early step in many research activities in the humanities is the collation and preparation of the material to be addressed. [64] This step may involve a variety of tasks, for multiple people, in dispersed locations. A typical research preparation phase in the humanities may involve an actual visit to an archive for a specific manuscript, digitisation of some selected pages, and eventually their transcription, mark-up, and annotation in a machine-readable format. Depending on the size of the project and the availability of the material, this process may take weeks or even years. [70] Consequently, task management is essential for many collaborative projects, and the particular challenge in this use case pertains to its distributed nature, which may involve multiple independent repository systems. A system supporting task management in distributed teams monitors changes to the material in its distinct sources (e.g. newly incoming digitisations, updates to transcriptions), and allows researchers to annotate the state of the material and to distribute tasks among team members.

Initiatives currently employ a variety of generic software packages [25]. Dedicated solutions are emerging for digitisation workflows [33] or as part of large editing systems [32]. However, we are not aware of any existing solution that spans multiple sources. To enable the construction of such systems in the first place, federation mechanisms are needed to read the metadata of available material from various sources, keep track of changes to those sources and material, and integrating the material (without necessarily extracting it from its original source).

Scenario C: Out-Sourcing Preservation Actions

The preservation of objects over long periods of time [14, 63] is a key challenge for repositories, in the face of the rapid advance of hard- and software environments. The importance of taking preservation actions has already shown in many spectacular cases, where important research data has been lost: up to 20 percent of the data of NASA's 1976 Viking mission to Mars have been lost [61]; satellite data recorded in the 1970s, which was to be used to identify ecological trends in South America's Amazon Basin, have been lost; and there are many more such negative examples (also in non-scientific contexts) [34, 44]. Trusted digital repositories [55] are assigned to reliably preserve their contents over time. Preservation of digital objects may involve strategies like migration, where files and metadata are transferred into newer or more stable formats, before old formats run danger of becoming obsolete. [71] This migration process – or "conversion" as the technical aspects of transferring an object into another format is called – may need to be conducted external to the repository for two reasons. [29] First of all, batch conversion – e.g. of all TIFF files to JPEG2000, or all PDF files to PDF/A [54] – may be compute-intense and hence it should not be conducted directly on the live repository server. At the same time, there may already be external services that offer conversion capabilities, and a repository that receives a myriad of different formats on ingest may not be in the place of providing dedicated conversion services for all of these formats. [52]

3. Technical Aspects

3.1. Federation Patterns

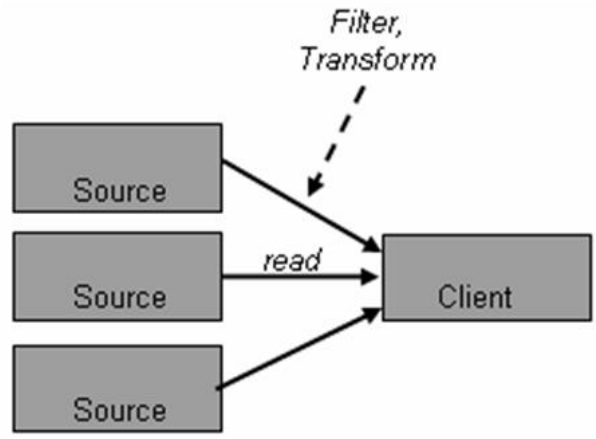
Repository federation encompasses viewing, re-using or processing both, individual objects as well as entire sets of objects, between independent software agents. The agents involved can be digital repositories or any other software agent in a repository environment (e.g. registries, added-value services).

Some of the challenges that are addressed by federation patterns to a different degree include

- **efficiency** – Efficiency in a federated environment is particularly dependent on the multiple, independent agents. Each additional agent raises the risk that the low performance of that one agent impacts detrimentally on the overall performance of the whole federation.
- **consistency, completeness** – As digital objects are duplicated and passed between independent agents, consistency issues may arise. Particularly in environments where objects change frequently, clients may hence be presented with old versions of an object or with processing results building on such old versions. Likewise, delays in the propagation of a newly added object through the federation may lead to an incomplete state at federated agents.
- **scalability** – The overall performance of a federation should not degrade with an increasing number of agents.
- **openness** – This thesis argues that openness is one of the key properties of federations. In particular, it characterises 'openness' to be constituted of the three attributes loosely-coupled, simple, and decentralised (see above).
- **standard** – Enabling openness and decentralisation indirectly calls for a minimum level of standardisation or also the flexibility to embed standards with regard to syntax, semantics, or structure, since standards support the implementation of federation mechanisms into decentralised agents that build on heterogeneous platforms and are governed independently.

3.1.1. Distributed Query

A Distributed Query essentially is the composition of multiple Client/Server interactions, as a query is sent to multiple sources and the responses are subsequently integrated into a single result set. The client must know all sources, and ideally the sources all provide a single standard interface for the query. Result sets can be filtered through adaptation of the query; responses can be transformed on delivery either through re-representation services or workflows.



Application Context: A Distributed Query pattern is best used in a setting where objects in the disparate sources may change frequently and at any time. At the same time, however, the client wants to access the very latest object versions, and consistency problems between the various sources need to be avoided. Another reason to opt for a Distributed Query pattern for repository federation may be technical constraints (e.g. large size) or legal restrictions, as the data remains at the source institution (other than in the case of Notification or Harvest patterns, see below).

Forces: Even with dedicated server interfaces, Distributed Queries are often difficult to integrate along both, efficiency and content at the same time. A Query is often dependent on the slowest server, when clients aim to integrate the various responses into a single result set. Thus, particularly in decentralised environments where clients have little influence on the source's quality of service, slow response times of some sources may be prohibitive for adequate results. Underlining this, the Resource Discovery Network (RDN) was finding that even with only "five subject gateways in its cross search there were problems of poor performance" [21].

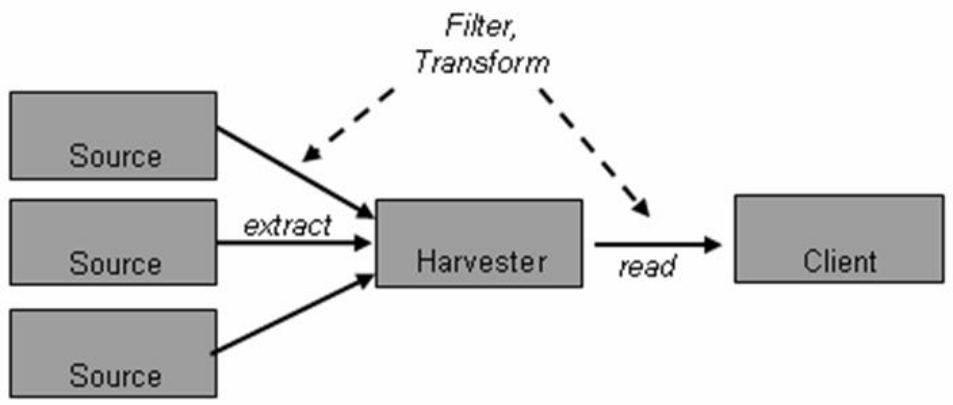
Exemplary Implementations: There are various implementations of the Distributed Query pattern in the repository community. Z39.50 for querying library catalogues has been around since 1988. Z39.50 was widely spread and still is, along with its successor SRU/W that is based on web services respectively REST. While Z39.50 and SRU/W merely exchange object metadata, other messages are conceivable including added-value services [48, 67].

One of the notable implementations in other communities is SDMX (see above), the protocol for Statistical Data and Metadata eXchange supports federations that may span numerous organisations around the globe. SDMX has been chosen, since statistical data are often subject to licenses and cannot be hosted outside of the creator's organisational environment. Another characteristic in the statistical data domain, that makes Distributed Query the suitable pattern, are the rigid consistency requirements in the face of frequent update cycles.

3.1.2. Harvest

An intermediary between source and client – the harvester – collects all the relevant data from disparate sources, and provides a single, integrated portal to the client. Regular harvest cycles ensure that the data gathered by the harvester remains up-to-date. The harvest mechanisms may amongst other vary as to how the sources are identified, how often harvest cycles are performed, and whether a follow-up harvest cycle only updates changed data (iterative) or recollects all the data regardless of whether or not it was updated (complete).

Filtering of the objects to be exchanged occurs in the communication between the source and the harvester, if the source provides relevant stubs. A transformation of objects can theoretically be conducted during the harvest, though we are not aware of any respective implementation in practice.



Application Context: The Harvest pattern de-couples the client from the server thereby scaling the communication in the federation down from multiple tiers to only two: the client and the harvester. This potentially improves the response time for clients considerably. Therefore, the Harvest pattern is suitable for decentralised environments, in which independent sources may not offer adequate quality of service with regard to their response time.

Furthermore, as is outlined in the next paragraph, the Harvest pattern is best used in environments where digital objects change infrequently due to the potential data inconsistencies introduced by the Harvester.

Forces: The redundant storage of data may introduce inconsistencies to the original, which is further aggravated through infrequent updates. Infrequent updates, in turn, may be enforced on the overall system as harvest cycles potentially take considerable time, depending on the size of the federation, server response, and the size and complexity of the digital objects involved. [11]

Pattern Details: Harvesters such as those for web search engines are well researched, and there are relevant experiences from this community. [36] However, there are some differences to harvesting mechanisms in repository environments that we will focus on in the following.

With regard to the potential inconsistencies and the load on the harvester, as mentioned above, the key mechanism is data selection: which object should be downloaded, and when? There must be a mechanism for identifying objects in the first place, and in the following we present three conceivable mechanisms.

- Web search engines usually follow-up the links parsed out of the harvested data, thereby establishing a self-referencing network of web resources. This is not feasible in repository environments, which mostly lack such densely linked content.

- In another approach, the server brokers the data to the harvester. In one way to achieve this, the server passes the ID of the next object along with a harvested resource (a "resumption token"). However, this either introduces state between the server and the client which potentially affects the robustness of the system, or it may lead to inconsistencies if the list of objects changes during the harvesting cycle. [49]
- In an alternative approach, the repository or other object source needs to provide a list of its objects. The way such a list is provided may vary from merely a plain list, to a list with details about when the object was last updated, to a dynamic list that can be queried for specific object attributes including last update. [12]

An additional impact on the overall efficiency of the system can be achieved by including information about the last update of an object and other metadata in the selection decision. Metadata about the last update may be useful, in case a harvester re-visits a source to only retrieve the objects that were updated since its last visit – iterative harvesting rather than complete harvesting rounds. More extensive filtering may be applied at this point of selection.

Exemplary Implementations: The Harvest pattern is well known in the repository community due to its implementation in OAI-PMH – probably the most prevalent federation mechanism today (see above). OAI-PMH is geared at harvesting purely metadata, not the actual content of an object. However, the protocol has been employed in various contexts (e.g. [24, 46, 56]) and it has also been tweaked to harvest whole objects marked up in METS or MPEG-DIDL [69]. One may argue though that these adaptations on OAI-PMH were mainly driven by the prevalence of OAI-PMH, not because OAI-PMH is really the most suitable technology for use cases other than metadata harvesting.

At the same time, we are not aware of any other significant implementation of the Harvest pattern The low occurrence of alternative harvesting mechanisms to OAI-PMH in repository environments notwithstanding, it is quite simple to implement the Harvest pattern ad hoc using other existing mechanisms. For example, "sitemaps" [57] offer the crawlers of web search engines a standard entry point to the contents of web sites, and it could equally be used to expose repository contents for harvesting by repository services. Sitemaps also offers a lastmod field that encodes the object's last modification date, to support iterative harvesting.

Taking this one step further unveils a connection between the Harvest and the Hybrid Notification pattern (see below). The Sitemaps exposing the repository contents (exemplary described above as an alternative mechanism for the Harvest pattern) are very similar to exposing repository contents via an Atom feed (an exemplary implementation suggested for the Hybrid Notification mechanism). Similarly, recurrent iterative harvesting cycles are comparable to polling the message queue. The only difference between the two patterns is that for the Harvest pattern a complete list of objects (and object metadata including the last update) is exposed, whereas Atom-feeds provide a history of repository events and hence can only infer the complete list by reconstructing the current state.

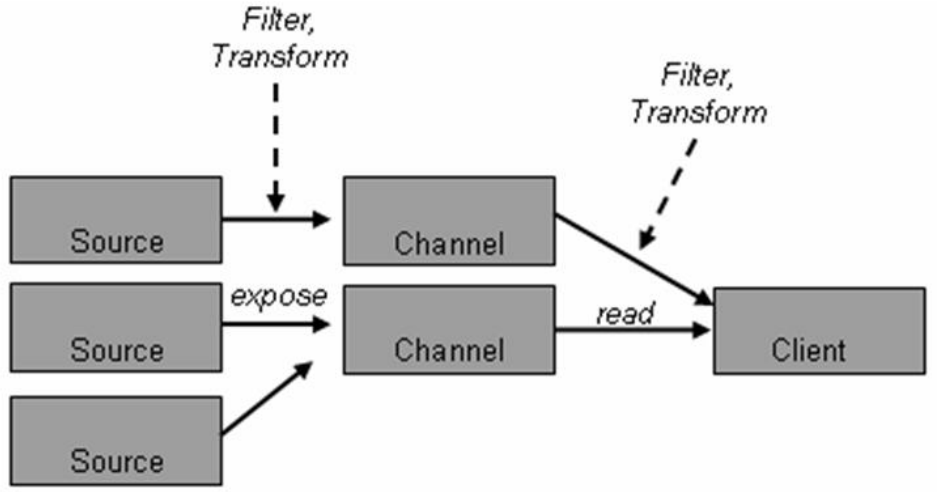
In conclusion, the Sitemaps-based harvesting shows that the Harvest pattern is a universal pattern that is not tied to OAI-PMH or any specific technology. Furthermore, the touching point between iterative harvesting and hybrid notification can be interpreted as an indicator of the completeness of the pattern language at this point.

3.1.3. Notification

In a Notification pattern, the source sends out messages on repository events. Triggers for notifications can be e.g. CrUD events – the creation, update, or deletion of an object in the repository –, which allows the client to stay in sync with the current state of the repository. A

common focus on CrUD events could facilitate a standard interface across heterogeneous agents, yet specialised notifications are conceivable.

We distinguish between two sub-patterns of Notification: Notification by Registration, and a Hybrid Push/Poll Notification, which are described below. Both build on the availability of a message channel, which conveys the notifications from the source to the client. Filters can be applied during the exposure into the channel respectively on read.



Application Context: Notification is particularly suited for federation topologies where the agents are synchronised in their state, and need information about repository events as they occur. Once many independent agents need to be synchronised, a Notification pattern is more timely than Harvest, and more robust than a Distributed Query pattern by its direct, yet decoupled communication between the source and the client. [39]

Forces: A Notification pattern requires the setup of a suitable message channel where messages are actively exposed by the source. Particularly in approaches that are by Registration, the reliability of this channel is of key importance. Also and particularly in a Hybrid approach, the latency of transporting the message from source to client must be taken into account.

Pattern Details: Notifications can be interpreted as the opposite of the Distributed Query mechanism. While in a Query the client requests information from a set of sources in a lower architectural layer, notifications are triggered by low-level events and passed on to higher level services. [19] The implementation of e.g. an Observer pattern on CrUD events allows the client to follow state changes in the repository as they occur. [18]

A Notification pattern builds on a message channel, and we distinguish broadly two approaches of how such a channel can be implemented. The first approach is “by Registration”, with some messaging frameworks distinguishing between publish-subscribe (one-to-many) and point-to-point (one-to-one) models. [40, 72]

Both messaging models require an event mechanism that allows subscription in the publish-subscribe model (which delivers immediately on the occurrence of an event), or the creation of a dedicated queue in the point-to-point model (which delivers on consumption, and hence reliably delivers messages). Because of the registration and since the notifications are passed on without delay, this pattern is often used in more tightly-coupled environments.

In contrast to these registration-based notifications, Hybrid push/poll notifications (many-to-many) can be initiated without any communication between the agents and are hence more decoupled. Instead of the subscription process or a dedicated queue, consumers retrieve notifications from a broker. This broker may offer a notification history, such that a client can

look up past notifications or it may be offline when a notification is sent and retrieve it later whenever convenient. This increased decoupling and robustness comes at the cost of immediacy, since the consumer needs to actively retrieve the notification. In the worst case a delay of a whole poll cycle is needed until a notification is retrieved. However, this impact is generally not seen as critical as pointed out e.g. by the cloud infrastructure provider Bycast [58]. Bycast's Hybrid push/poll notification system is at the core of its cloud infrastructure, and as a mechanism for its broker, it employs the Atom syndication protocol.

Exemplary Implementations: Few repositories have adopted message-oriented middleware for coordinating repository-internal processes. Since version 3.0, Fedora implements the Java Messaging Service JMS [30]. Fedora sends notifications on all calls to its API-M, which includes CRUD operations on objects, datastreams, and relations. At the time of writing, DSpace is preparing a new event system for the release of its version 2.0. [50]

The probably most comprehensive implementation of messaging is in place in the iRODS rules system that is triggered through administrative actions. [51] The iRODS rule system provides a customizable framework for executing tasks – so-called "microservices" – on occurrence of definable events. In a way, rules register microservices with specific events, and because of this basic similarity we can classify rules as event-based notifications. However, rules go beyond notifications since they are capable of defining microservice workflows. [16] All these messaging frameworks existing in repository installations, however, are system-internal. We are not aware of an open approach that is employed as a federation mechanism across heterogeneous agents in a repository environment. A first step towards such a Notification-based Federation could be a Hybrid Notification based on the Atom protocol. Since Atom is an XML-based standard, it enables communication across heterogeneous agents with different software bases. As mentioned above, implementations of a Hybrid Notification pattern are considered viable even in a multi-agent environment where timing is an issue and hence frequent poll-cycles are required. [45, 58] Its embedding into the web architecture may be conducive to this, as conditional HTTP GET requests and common caching mechanisms in web proxies minimise the impact of short polling cycles by consumers.

Yet, such an Atom-based Hybrid Notification pattern remains to be tested in a repository environment. In the following section we describe a prototype for that, which was developed for TextGrid in the context of the DARIAH e-Humanities infrastructure.

3.2. An Atom-based Repository Federation

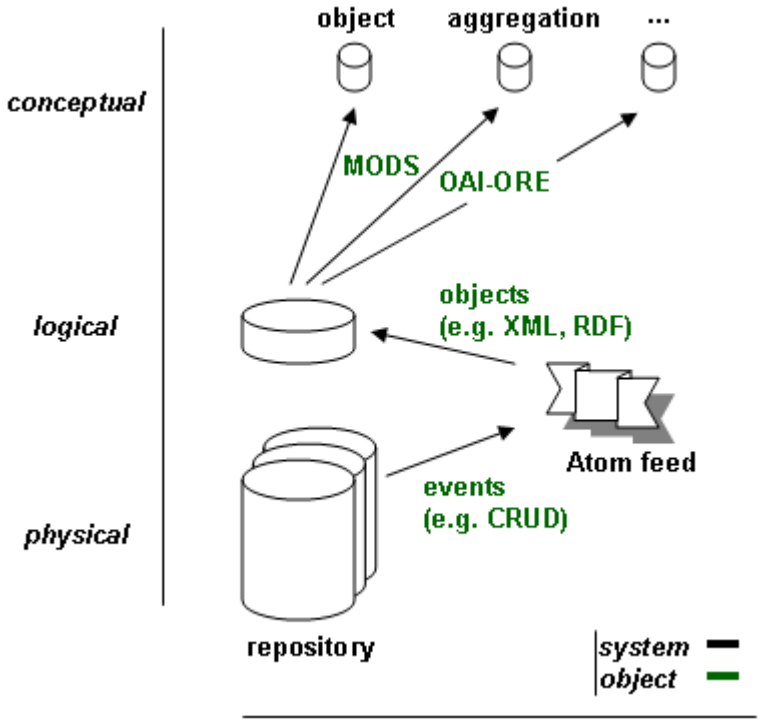
After the previous sections introduced the context and concept of Federation patterns, this section presents an actual federation environment with multiple repositories and other independent agents. This environment will establish DARIAH, a closely-knit, yet open repository infrastructure for the humanities. [4, 10] The close interaction between the heterogeneous agents calls for a Notification-based federation approach. Therefore, this section illustrates the application of Atom-based Hybrid Notification.

DARIAH is a project in the framework of the European Strategy Forum on Research Infrastructures (ESFRI) [28] and is currently in its initial phase. ESFRI projects are designed to offer research communities essential infrastructure for decades to come, e.g. a large telescope for astronomy and an icebreaker ship for the polar sciences. For the humanities, DARIAH builds a digital infrastructure to share cultural artefacts, re-use existing tools, and collaborate across institutional, cultural, and disciplinary boundaries. Partners in DARIAH include researchers and humanities centres, including DANS (Data Archiving and Networked Services)

in the Netherlands, the Centre for e-Research (CeRCH) at King's College London, as well as the State and University Library Goettingen, Germany.

The goals for the DARIAH repository infrastructure lie particularly in the combination of two characteristics: the repositories should remain independent, grow and evolve over time, and interact with other agents (hence open), while the contents in DARIAH and functionalities provided through DARIAH should be accessible to the researcher as if DARIAH was a single platform (hence closely-knit). Foremost, as a virtual research environment that supports active research, resources in DARIAH may change over time and in an early stage of creation they may indeed be private. These prerequisites – decentralised, heterogeneous agents that need to stay in sync with the state of other agents; with digital objects that may change frequently – call for a Notification-based approach.

The DARIAH test environment for linking heterogeneous repositories spans three different systems: TextGrid, iRODS, and Fedora. In order to synchronise the states of the three repositories, notifications are sent to the respective other repositories on the creation or modification of a digital object. In the production environment this mechanism will be used to replicate data across multiple sites, and to update external applications such as search and analysis about changes made in any of the DARIAH sites. Both, iRODS and Fedora offer internal event mechanisms – iRODS through its rules/microservices [51], and Fedora implements the Java Message Service JMS [30].



The exposure of repository events via Atom can be directly integrated into these event mechanisms. While TextGrid does not offer an internal event mechanism, the TG-crud interface handles all updates to objects in TextGrid and it can easily be adapted to expose object creation or modification. The Atom feeds from the three repositories are all handled via a single Apache Abdera[14] server. While the production environment will likely consist of multiple Atom servers, a central server is sufficient for the test environment.

[14] http://abdera.apache.org/

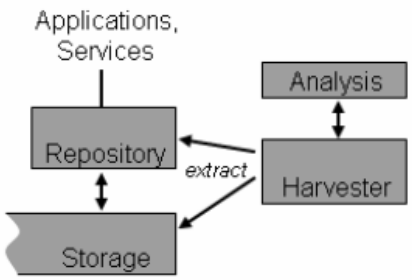
The repositories poll the feeds of the respective other servers and thereby synchronise with their states. As an additional feature, a repository may offer multiple feeds via the Atom server – e.g. one feed exposes all the objects, whereas others may only expose specific format types such as only XML objects. This type of server-side filtering is more efficient for both client and server – for the client since it does not need to filter itself based on the metadata of the object, for the server since this will reduce overall polling.

The reason for why this will reduce overall polling at the server is related to the fact that Atom feeds are HTTP-based services, embedded in the web architecture, and hence also supported by the infrastructure of proxies and caching servers. Furthermore, polling an Atom feed that is unchanged only puts minimal load on the Atom server.

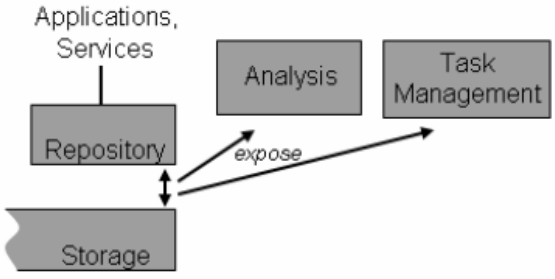
Specifically, conditional HTTP GET's (i.e. the HTTP header 'If-Modified-Since') ensure on a HTTP level that the feed is only downloaded if it actually changed. In conclusion, we have presented the DARIAH research infrastructure for the humanities, the diversity of its collections and the vision of an open environment of decentralised agents. To ensure coherence among these decentralised agents as well as in communication with related initiatives, the DARIAH federation builds on an Atom-based notification pattern as one of its key design ideas. An experimental setup that links TextGrid, an iRODS and a Fedora test server have demonstrated the viability of this approach.

But how do these results relate to the federation scenarios put forth in section 2 – specifically Scientific Analysis and Task Management?

Search and Analysis is a recurrent requirement in the DARIAH environment. However, a simple Google-type search is insufficient for a scientific environment. Specialised analysis services may process various types of data and their metadata, including images and sound. In other words, rather than providing a generic search portal, DARIAH aims to facilitate the creation of external search and analysis services, such that any community or project can develop their own portal. Thereby, one-time analysis efforts that research a specific question on a specific set of digital objects are offered possibilities to harvest the objects into a dedicated analysis environment. Ongoing services that grow with the availability of new material are provided with notifications about object creation, update, or deletion. The infrastructure therefore has to support various protocols and patterns, to allow for different approaches to system interoperability.



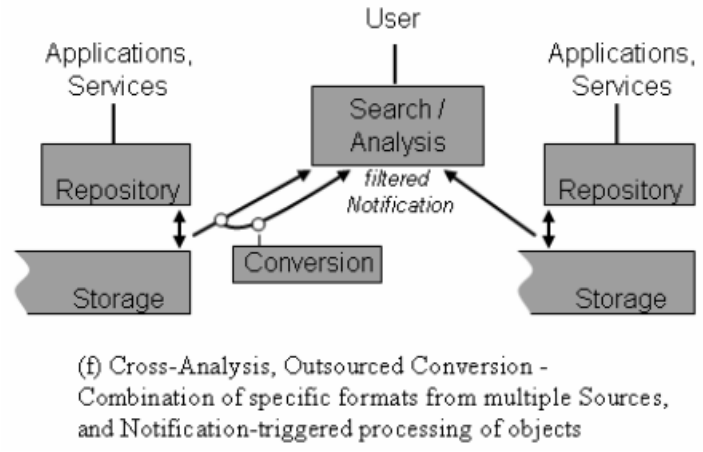
(d) Scientific Analysis -
Objects extracted through Harvest,
for one-time, bulk Analysis



(e) Scientific Analysis, Task Management -
Objects exposed through Notification,
for establishing a synchronised, ongoing service

Just like this infrastructure aims to support various approaches to system interoperability, it equally aims to expose its content in different formats to facilitate different approaches to object interoperability. To underline this, the Task Management scenario is similar to the search and analysis portals described above, yet it operates purely on the object metadata and whether objects are available in the first place. Thereby, the exposure of object metadata

through a Query or a Notification pattern, as well as adequate filters enable the implementation of a Task Management application.



(f) Cross-Analysis, Outsourced Conversion - Combination of specific formats from multiple Sources, and Notification-triggered processing of objects

DARIAH aims to foster interoperability of these mechanisms across the diverse repositories and other agents in the DARIAH infrastructure – like TextGrid services or the TextGrid Repository. This allows that external agents can embed their own application environments into this infrastructure, just like it enables the Scientific Analysis and the Task management scenarios.

4. Next Steps

Decentralised information environments are emerging, in which a repository is but one agent among a multitude of others. To name just some of the conceivable scenarios of such environments, repositories may replicate relevant objects of another source (e.g. institutional vs. thematic repositories), parts of a single digital object may be spread over various repositories (e.g. e-Publications with DRIVER[15]), repositories may depend upon external re-representation and preservation services [73].

Rather than convergence to a small set of concepts and technologies, we are expecting diversity and decentralisation to increase in repository federations. New application contexts of repositories (e.g. data-driven research, enterprise systems) and subsequently changing requirements to repository infrastructure, as well as the ongoing integration of new technologies (e.g. Linked Data[16], clouds as in DuraCloud[17]) in the field seem to point that way. In the face of this growth and diversity, the approach presented in this paper may contribute to a more structured discussion and avoid disintegration and redundancies within the repository community

The design of TextGrid is grounded in the organisational and social context of research in the humanities. The principles – generic infrastructure, specialised functionalities, and participation – can be mapped onto the current TextGrid architecture or open repository environments in general. As a factor for encouraging participation and lowering entry barriers, layered approaches put themselves forward also for organisational aspects:

A layered approach manifests itself in various aspects, including data, services and preservation stores. To illustrate the TextGrid collaboration layers on data:

---

[15] http://www.driver-repository.eu/Enhanced-Publications.html

[16] http://linkeddata.org/

[17] http://www.duraspace.org/duracloud.php

- any data format can be uploaded, TextGrid ensures bit-preservation
- metadata facilitates data management and retrieval (metadata-based search)
- by uploadig XML-based texts, a series of services can be used on the data including streaming tools, an XML-editor, and other functionalities
- if the XML follows TEI encoding, TextGrid offers graphical editing, metadata extraction, and other functionalities
- defining a mapping to the TextGrid recommendation for a TEI baseline encoding allows interoperability on a semantic level

In its design of the incentive approach, TextGrid follows the experiences of collaborative environments. [74] The user can do whatever she wants, but by being interoperable and compliant to TextGrid recommendations she increases exposure and is provided with more functionality in the TextGrid virtual research environment.

Layered conventions are e.g. conceivable for both data and service interoperability, reaching from low interoperability with a low entry barrier to high interoperability and hence a high value for re-use and collaboration. TextGrid supports the creation of such layered community conventions with the same mechanisms with which it supports the fusion of generic infrastructure and specialised functionalities in a single environment.

While the robust flexibility of TextGrid facilitates all these principles, they are essentially social and organisational notions rather than technical ones. In other words, while the technology is ready to support what is needed, this openness requires a higher level of organisation within the user community. Other open repository environments may therefore opt to constrict the openness to suit the community. A key role in this organisational process may be played by the Confederation of Open Access Repositories (COAR)[18].

5. References

[1] Chris Anderson, The End of Theory: The Data Deluge Makes the Scientific Method Obsolete, WIRED MAGAZINE: 16.07. http://www.wired.com/science/discoveries/magazine/16-07/pb_theory

[2] ANDS Technical Working Group. Towards the Australian Data Commons – A proposal for an Australian National Data Service. Australian Government, Department of Education, Science and Training, October 2007. http://www.pfc.org.au//pub/Main/Data/TowardstheAustralianDataCommons.pdf.

[3] Andreas Aschenbrenner, Reference Framework for Distributed Repositories – Towards an Open Repository Environment (PhD Thesis, Göttingen 2010), http://resolver.sub.uni-goettingen.de/purl/?webdoc-2390

[4] Andreas Aschenbrenner. e-Humanities in Europe – DARIAH. In Proceedings of the Open Grid Forum 20, Manchester, UK, May 7-11 2007.

[5] Andreas Aschenbrenner et al., "Open ehumanities digital ecosystems and the role of resource registries," in 2009 3rd IEEE International Conference on Digital Ecosystems and Technologies (gehalten auf der 2009 3rd IEEE International Conference on Digital Ecosystems and Technologies (DEST), Istanbul, Turkey, 2009), 745-750. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5276672.

[18] http://coar-repositories.org/

[6] Andreas Aschenbrenner, Tobias Blanke, Marc W. Küster, Wolfgang Pempe, "Towards an Open Repository Environment," Journal of Digital Information, Vol 11, No 1 (2010), http://journals.tdl.org/jodi/article/view/758

[7] Andreas Aschenbrenner, Tobias Blanke, Neil P Chue Hong, Nicholas Ferguson, and Mark Hedges. A Workshop Series for Grid/Repository Integration. D-Lib Magazine, 15(1/2), January/February 2009.

[8] Andreas Aschenbrenner, Tobias Blanke, Stuart Dunn, Martina Kerzel, Andrea Rapp, and Andrea Zielinski. Von e-Science zu e-Humanities - Digital vernetzte Wissenschaft als neuer Arbeits- und Kreativbereich für Kunst und Kultur. Bibliothek, Forschung und Praxis, 31(1), 2007. http://www.bibliothek-saur.de/2007_1/011-021.pdf.

[9] Andreas Aschenbrenner, Tobias Blanke, David Flanders, Mark Hedges, and Ben O'Steen. The Future of Repositories? - Patterns for (Cross-)Repository Architectures. D-Lib Magazine, 14(11/12), November/December 2008.

[10] Andreas Aschenbrenner, Tobias Blanke, Eric Haswell, and Mark Hedges. The DARIAH e-Infrastructure. Zero-In Magazin, 3, October 2009.

[11] Andreas Aschenbrenner and Andreas Rauber. Die Bewahrung unserer Online-Kultur. Vorschläge zu Strategien der Webarchivierung. Sichtungen, 2003.

[12] Automatisiertes Abliefern über Harvesting-Verfahren – Wege zur effizienten Ablieferung von Netzpublikationen. Deutsche Nationalbibliothek, August 2008. http://www.d-nb.de/netzpub/ablief/pdf/automatisierte_ablieferung.pdf

[13] Berlin Declaration on Open Access to Knowledge in the Sciences and Humanities, October 2003. http://oa.mpg.de/openaccess-berlin/berlindeclaration.html.

[14] Francine Berman. Got data?: A guide to data preservation in the information age. Commun. ACM, 51(12):50–56, 2008.

[15] Tobias Blanke, Andreas Aschenbrenner, Marc Küster, Christoph Ludwig: No Claims for Universal Solutions - Possible Lessons from Current e-Humanities Practices in Germany and the UK. In: e-Humanities - An Emerging Discipline. Workshop at the 4th IEEE International Conference on e-Science. December 2008. http://www.clarin.eu/system/files/ClaimsUniversal-eHum2008.pdf

[16] Tobias Blanke and Mark Hedges. Providing linked-up access to Cultural Heritage Data. In Proceedings of the ECDL 2008 Workshop Information Access to Cultural Heritage (IACH), Aarhus, Denmark, September 18 2008. http://ilps.science.uva.nl/IACH2008/proceedings/proceedings.html.

[17] Martha L. Brogan. Contexts and Contributions: Building the Distributed Library. Technical report, University of Pennsylvania, 2006. http://repository.upenn.edu/librarypapers/31.

[18] Frank Buschmann, Kevlin Henney, and Douglas C. Schmidt. Pattern-Oriented Software Architecture – A Pattern Language for Distributed Computing, Volume 4 of Software Design Patterns. John Wiley & Sons Ltd., 2007.

[19] Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, and Michael Stal. Pattern-Oriented Software Architecture, Volume 1: A System of Patterns. John Wiley & Sons, August 1996.

[20] Priscilla Caplan, William Kehoe, Joseph Pawletko , International Journal of Digital Curation, Vol 5, No 1 (2010). http://www.ijdc.net/index.php/ijdc/article/view/145

[21] Leona Carpenter. OAI for Beginners – the Open Archives Forum online tutorial, 2003. http://www.oaforum.org/tutorial/english/intro.htm.

[22] Annamaria Carusi and Torsten Reimer, “Virtual Research Environment Collaborative Landscape Study.” Report issued by the JISC in January 2010. http://www.jisc.ac.uk/publications/reports/2010/vrelandscapestudy.aspx

[23] Elizabeth Chang and Marc Wilhelm Küster, “IEEE-DEST 2011 ----- Daejeon, Korea,” IEEE DEST: Background and Objectives, Juni 2010, http://dest2011.debii.curtin.edu.au/.

[24] Churngwei Chu, Walter E. Baskin, Juliet Z. Pao, and Michael L. Nelson. OAI-PMH architecture for the nasa langley research center atmospheric science data center. In Julio Gonzalo, Costantino Thanos, M. Felisa Verdejo, and Rafael C. Carrasco, editors, Proceedings of the ECDL 2006, volume 4172 of Lecture Notes in Computer Science, pages 524–527. Springer, 2006.

[25] Comparison of project management software. Wikipedia entry, Viewed August 2009. http://en.wikipedia.org/wiki/Comparison_of_project_management_software.

[26] Tharam S. Dillon, Chen Wu, and Elizabeth Chang. Reference Architectural Styles for Service-Oriented Computing . In Network and Parallel Computing. Proceedings of the IFIP International Conference NPC 2007, volume 4672 of Lecture Notes in Computer Science, pages 543–555, Dalian, China, September 2007. Springer.

[27] Stuart Dunn, (2009) "Dealing with the complexity deluge: VREs in the arts and humanities", Library Hi Tech, Vol. 27 Iss: 2, pp.205 – 216. http://dx.doi.org/10.1108/07378830910968164

[28] European Strategy Forum on Research Infrastructures (ESFRI). European Roadmap on Research Infrastructures, 2006. http://cordis.europa.eu/esfri/roadmap.htm.

[29] Adam Farquhar and Helen Hockx-Yu. Planets: Integrated Services for Digital Preservation. International Journal of Digital Curation, 2(3), 2007.

[30] Fedora Messaging Guide. Fedora Commons Report, Viewed August 2009. http://www.fedora-commons.org/documentation/3.0/userdocs/server/messaging/index.html.

[31] Roy Thomas Fielding. Architectural Styles and the Design of Network-based Software Architectures. PhD thesis, University of California, Irvine, 2000.

[32] Forschungsnetzwerk und Datenbanksystem (FuD). Project Website, Viewed August 2009. http://fud.uni-trier.de/.

[33] Goobi – Digital Library Modules. Project Website, Viewed August 2009. http://goobi.sub.uni-goettingen.de/.

[34] Richard Harada. Are you prepared for long-term data preservation? - first in/first out. Computer Technology Review, October 2003. http://findarticles.com/p/articles/mi_m0BRZ/is_10_23/ai_111062977/.

[36] Erik Hatcher and Otis Gospodnetic. Lucene in Action. Manning Publications, Decem-ber 2004.

[37] Tony Hey, Stewart Tansley, Kristin Tolle (eds.) The Fourth Paradigm: Data-Intensive Scientific Discovery, Microsoft Research,Redmond, Washington, 2009.

[39] Gregor Hohpe. Programming Without a Call Stack – Event-driven Architectures. Whitepaper, 2006. http://www.enterpriseintegrationpatterns.com/docs/EDA.pdf.

[40] Gregor Hohpe. SOA Patterns - New Insights or Recycled Knowledge? Whitepaper, May 2007. http://www.eaipatterns.com/docs/SoaPatterns.pdf.

[41] Gregor Hohpe and Bobby Woolf. Enterprise Integration Patterns – Designing, Building, and Deploying Messaging Solutions. Addison-Wesley, December 2008.

[42] JISC Common Repository Interfaces Group (CRIG). website, Viewed August 2009. http://www.ukoln.ac.uk/repositories/digirep/index/CRIG.

[43] Marc Wilhelm Küster, Christoph Ludwig, and Andreas Aschenbrenner, "TextGrid as a Digital Ecosystem," in DEST 2007, hg. v. Elizabeth Chang, 2007. http://www.textgrid.de/fileadmin/TextGrid/veroeffentlichungen/DigitalEcosystem07_CameraReady-1.pdf

[44] Leo Lewis. Scandal over lost pensions may be the final straw for ruling party. The Times, July 3 2007. http://www.timesonline.co.uk/tol/news/world/asia/article2017410.ece.

[45] Heiko Ludwig, Jim Laredo, Kamal Bhattacharya, Liliana Pasquale, and Bruno Wassermann. REST-Based Management of Loosely Coupled Services. In Proceedings of the International World Wide Web Conference (WWW2009), Madrid, Spain, April 20-24 2009.

[46] Liz Lyon, Rachel Heery, Monica Duke, Simon J. Coles, Jeremy G. Frey, Michael B. Hursthouse, Leslie A. Carr, and Christopher J. Gutteridge. eBank UK: linking research data, scholarly communication and learning. In Proceedings of the UK e-Science All Hands Conference, pages 711–719. Engineering and Physical Sciences Research Council, 2004.

[47] National Science Foundation (NSF). Sustainable Digital Data Preservation and Access Network Partners (DataNet). Funding Call August 2007. http://www.nsf.gov/funding/pgm_summ.jsp?pims_id=503141.

[48] Names: Pilot national name and factual authority service. JISC Project Outline, Viewed August 2009. http://www.jisc.ac.uk/whatwedo/programmes/reppres/sharedservices/names.aspx

[49] Cesare Pautasso and Erik Wilde. Why is the Web Loosely Coupled? A Multi-Faceted Metric for Service Design. In Proceedings of the 18th International World Wide Web Conference, pages 911–920, Madrid, Spain, April 2009. ACM Press.

[50] Prototype Implementation of New Event System. DSpace Development Portal, Viewed August 2009. http://wiki.dspace.org/index.php/EventSystemPrototype.

[51] Arcot Rajasekar, Mike Wan, Reagan Moore, and Wayne Schroeder. A Prototype Rule-based Distributed Data Management System. In Proceedings of the HPDC workshop on "Next Generation Distributed Data Management", Paris, France, May 2006.

[52] Jose Carlos Ramalho, Miguel Ferreira, Luis Faria, Rui Castro, Francisco Barbedo, and Luis Corujo. RODA and CRiB a service-oriented digital repository. In Proceedings of the International Conference on Preservation of Digital Objects (iPRES), London, 2008.

[53] Andreas Rauber, Andreas Aschenbrenner. Web Archiving, chapter Mining Web Collections, pages 153 – 176. Springer, 2006.

[54] Carl Rauch, Harald Krottmaier, and Klaus Tochtermann. File-Formats for Preservation: Evaluating the Long-Term Stability of File-Formats. In Proceedings of the ELPUB2007 Conference on Electronic Publishing, Vienna, Austria, June 2007.

[55] Research Libraries Group and OCLC. Trusted Digital Repositories: Attributes and Responsibilities, 2002.

[56] Uwe Schindler, Benny Bräuer, and Michael Diepenbroek. Data information service based on open archives initiative protocols and apache lucene. In Proceedings of the German e-Science Conference (GES), Baden-Baden, Germany, 2007. Max-Planck Society.

[57] Sitemaps XML format. Format Specification, February 2008. http://www.sitemaps.org/protocol.php.

[58] David Slik. Bycast's Cloud Storage HTTP API. Presented at the SNIA Cloud Storage Group Meeting, May 2009. http://groups.google.com/group/snia-cloud/web/cloud-storage-twg-chicago-2009.

[59] Statistical Data and Metadata Exchange Initiative. SDMX Guidelines for the Use of Web Services, November 2005. http://www.sdmx.org/docs/2_0/SDMX_2_0$%$20SECTION_07_WebServicesGuidelines.pdf.

[60] Statistical Data and Metadata Exchange Initiative. SDMX User Guide, January 2007. http://sdmx.org/docs/2007/Conf07/doc%2031%20Capacity%20Building%20Room%20Document%20-%20UserGuide%20-%20Working%20Draft.doc.

[61] Marcia Stepanek. Data storage: From digits to dust. Business Week, April 20 1998. http://www.businessweek.com/archives/1998/b3574124.arc.htm.

[62] Robert Tansley. Building a Distributed, Standards-based Repository Federation – The China Digital Museum Project. D-Lib Magazine, 12(7/8), July/August 2006.

[63] Task Force on Archiving of Digital Information. Preserving digital information. Commissioned by the Commission on Preservation and Access and the Research Libraries Group, May 1996

[64] TextGrid. Scenarios. TextGrid Report, December 2006. http://www.textgrid.de/berichte.html.

[65] TextGrid. Text Retrieval. TextGrid Report 1.3, May 2007. http://www.textgrid.de/berichte.html.

[66] Kenneth Thibodeau. Overview of Technological Approaches to Digital Preservation and Challenges in Coming Years. CLIR Report 107, 2002. http://www.clir.org/pubs/reports/pub107/thibodeau.html.

[67] Barbara B. Tillett. A Virtual International Authority File. In Workshop on Authority Control among Chinese, Korean and Japanese Languages (CJK Authority 3), Karuizawa, Tokyo, Kyoto, March 14-18 2002.

[68] Towards a European e-Infrastructure for e-Science Digital Repositories. Presentation at the e-IRG Workshop, Lisbon, October 2007. http://oldsite.e-irg.eu/meetings/2007-PT/4-e_IRG_Pres_Oct07_v3.pdf.

[69] Herbert Van de Sompel, Michael L. Nelson, Carl Lagoze, and Simeon Warner. Resource Harvesting within the OAI-PMH Framework. D-Lib Magazine, 10(12), December 2004.

[70] Virtual Vellum. Final Report. JISC Project, February 27 2007. http://www.ahessc.ac.uk/files/active/0/VV-report.pdf.

[71] Colin Webb. Guidelines for the Preservation of Digital Heritage. UNESCO Report, United Nations Educational, Scientific and Cultural Organization, Paris, March 2003. http://unesdoc.unesco.org/images/0013/001300/130071e.pdf.

[72] John Wetherill. Messaging Systems and the Java Message Service (JMS). SUN Developer Network, Viewed August 2009. http://java.sun.com/developer/technicalArticles/Networking/messaging/.

[73] Steve Hitchcock, Tim Brody, Jessie M.N. Hey, and Leslie Carr. Digital Preservation Service Provider Models for Institutional Repositories – Towards Distributed Services. D-Lib Magazine, 13(5/6), May/June 2007.

[74] Amy Jo Kim. Community Building on the Web – Secret Strategies for Successful Online Communities. Peachpit Press, 2000.