A protocol for machine-readable cache policies in OGC web services: application to the EuroGeoSource information system

Rubén Béjar, Francisco J. Lopez-Pellicer, Javier Nogueras-Iso, F. Javier Zarazaga-Soria, Pedro R. Muro-Medrano

Computer Science and Systems Engineering Department, Universidad de Zaragoza, c/ María de Luna 1, 50018, Zaragoza, Spain

A protocol for machine-readable cache policies in OGC web services: application to the EuroGeoSource information system

Rubén Béjar, Francisco J. Lopez-Pellicer, Javier Nogueras-Iso, F. Javier Zarazaga-Soria, Pedro R. Muro-Medrano

Computer Science and Systems Engineering Department, Universidad de Zaragoza, c/ María de Luna 1, 50018, Zaragoza, Spain

---

Abstract

An efficient access to the contents provided through OGC web services, widely used in environmental information systems, is usually achieved by means of caching strategies. Service-owners may be interested in expressing the conditions required to allow for this. If these conditions are expressed in a machine-readable way, automatic harvesters can be programmed to follow them when caching those services.

This paper proposes a protocol to specify and follow cache policies for OGC web services expressed in a machine-readable language. A preliminary implementation of this protocol has been tested in the EuroGeoSource project, where a number of Web Feature Services providing mineral deposits and energy resources are periodically harvested and cached to improve the efficiency and availability of several applications. The protocol addresses a nowadays common case, and can possibly be extended to allow for more detailed policies. Further work will help to determine how it could be integrated into a full Digital Rights Management system.

Keywords: OGC web service, Rights Expression Language, Digital Rights Management, Cache, Energy, Mineral Deposit

---

1. Introduction

Geospatial services are widely used in environmental modelling software, as they provide a flexible, reusable alternative to monolithic and closed applications, and provide support for next-generation integrated environmental modelling systems (Granell et al., 2013; Laniak et al., 2013). Services following ISO/OGC (International Organization for Standardization / Open Geospatial Consortium) standards (we will refer to them as OGC web services) are being used to implement different computational models (e.g. alpine runoff events (Granell et al., 2010)), to share environmental data (e.g. historical landslides and floods (Salvati et al., 2009)), in environmental e-government applications (Latre et al., 2013), and also in combination with other services (Gebhardt et al., 2010; Peckham and Goodall, 2013).

Many types of OGC web services, such as Web Map Services (WMS), Web Map Tile Services (WMTS), Web Feature Services (WFS), Web Coverage Services (WCS) and Catalog Services for the Web (CSW), are "geographic model / information management services" according to the ISO 19119:2005 geographic services taxonomy (ISO, 2005). To access this kind of services with good performance, or to improve its availability, caching their contents is a good strategy. The most common practices are CSW metadata harvesting (Li et al., 2011; Deng and Wu, 2010) and WMS tiling (Liu and Nie, 2010; García et al., 2013). Tiling can be used for raster data (Lia et al., 2011), so it can be adapted to WCS. Caching is also starting to be applied to WFS vector data (Pla and Lleopart, 2010).

Caching is a technique often used in environmental software when performance is required, specially in distributed web-based systems and, more recently, in cloud based systems. For example, in a virtual database for ecological data, Frehner and Brändli (2006) use caching in their integration layer, on top of several WFSs. In another web services-based system, in this case for hydrologic data, Ames et al. (2012) use caches for keeping local copies of observational data series.

In the context of geospatial services, a *cache* often means a temporary storage for some of the contents offered by these services. In general terms, a cache improves the performance perceived by the users of a service because it allows to pre-generate, and reuse, the results produced by certain time-consuming operations offered by this service. The provider of the service may choose to establish a cache, but the users of this service may also choose to do so on their own side, i.e. in their own desktop computer or in a local server, if they need to, what is the more relevant case for this paper.

Caching a service can be a heavy load for it. For instance, tiling a WMS means making many thousands, even millions, of requests which must be added to those made by its regular users. Besides this, the cached contents, e.g. the map tiles, are stored for an undefined amount of time beyond the control of the service rights-holder. These are good reasons that may lead to establish and express certain conditions to cache a service contents. For instance, the Spain Cadastre WMS[1] can be used freely, but includes in its capabilities the prohibition to make tiled requests and massive downloads. The UK Ordnance Survey Open Space developer agreement[2] grants permission to the automatic, immediate and **temporary** storing (caching) of data. In France Géoportail, caching is prohibited unless an explicit license is obtained.[3]

As the contents provided through OGC web services change, caching those contents needs to be done periodically to keep them updated. Since the conditions to cache those services can change too, it would be useful to express them in a machine-readable way, so that a cooperating, automatic cache updater could react to those changes. However, this is not the current situation as the natural language licenses in the examples of the previous paragraph show.

This paper proposes a protocol to specify cache policies forOGC web services in a machine-readable Rights Expression Language (REL) that can be followed by cooperative *harvesters*. We will be using the term *harvesters* for automatic processes that cache the contents of geographic model/information management services. We need those harvesters to be "cooperative" because the protocol is not a full rights management system so it does not enforce the cache policies. The protocol is applicable to a nowadays common situation, and can also be a first step towards a Digital Rights Management (DRM) framework for those interested.

A preliminary version of this protocol has been tested in the EuroGeoSource project, see section 4, where a number of Web Feature Services (WFS) providing data on mineral deposits and energy resources are periodically harvested and cached in a central node to improve the efficiency and availability of several applications. The data provided through these services can be used as an input in environmental models like those proposed by Côte et al. (2010) or González et al. (2011).

The rest of this paper is organized as follows. The next section reviews work related to RELs and licensing in the geospatial web. Section 3 details

[1]http://ovc.catastro.meh.es/Cartografia/WMS/ServidorWMS.aspx?request=GetCapabilities&service=WMS
[2]`http://www.ordnancesurvey.co.uk/oswebsite/web-services/os-openspace/developer-agreement.html`
[3]`http://www.geoportail.gouv.fr/depot/api/cgu/licAPI_CGUF.pdf`

a protocol for OGC web services cache policies, based on ODRL 2.0. This protocol also establishes how to embed these policies in OGC web services (Section 3.3), and proposes an algorithm for cooperating harvesters to follow the policies (Section 3.4). Section 4 describes the EuroGeoSource project and how a first version of the protocol proposed in this paper was implemented there. Section 5 discusses the rationale behind some of the most significant decisions taken. To finish this paper, Section 6 summarizes the main conclusions and proposes some future lines of work.

2. Related work

Explicit license terms are necessary for geospatial assets (e.g. data and services) if the rights and obligations of their users must be clear. For instance, Spatial Data Infrastructures (SDIs) deal with this issue by defining more or less formal "Access Policies" (Béjar et al., 2012, p. 267) for their shared assets. Even open data and content[4] are not really open unless their license terms, or the legal conditions under which they are available, are well-known.

In the case of the environmental community, there is a clear interest in expressing use rights and data sharing and access policies since, for instance, they contribute to maximize the value of ecological data (Fegraus et al., 2005) specially when considering long term repositories (Michener et al., 2011). Regarding this kind of repositories, Jones et al. (2008) anlayze important issues about the "fair use of data", that include the right of the owners to determine who can access them. These issues are also considered important in biodiversity data repositories, although recent studies have found out that less than 40% of them actually do something (Bach et al., 2012). Lotz et al. (2012) consider property rights as key characteristics of curation and preservation when comparing biodiversity research databases, and recommend them as mandatory features for collaborative research projects. With regards to property rights, Janßen et al. (2011) provide an example of the necessity to take into consideration the intellectual property rights of data owners, in order to implement a vegetation database.

Free text licenses allow to express any obligations, permissions and prohibitions. These are not machine-readable, and they make no further attempt

[4] `http://opendefinition.org/okd/`

to verify that users have actually read or understood them. Click-through licenses are not machine-readable either, but they explicitly require the users to accept that they have "read and understood" the license before being granted permission to access the assets. The German GeoBusiness Commission has tried this approach for spatial datasets (Behrens and Reichling, 2011), as well as the Centre of Ecology and Hydrology in the UK (Vodden and Gartmann, 2011). This kind of licenses are adequate when human users are interactively involved in the process, but they do not try to address situations where computer systems (e.g. harvesters) are accessing the assets.

A rights expression language (REL) is a machine-readable language intended to express use rights and licenses for digital contents. RELs can be used as part of a DRM system, an access control technology used to limit the allowed use of digital contents, but the use of a REL does not imply that DRM is being used. A widely successful example of the use of a REL without DRM in the World Wide Web is the robot exclusion protocol[5], which is a consensus-based standard that uses a simple expression of machine-readable rights to express which parts of a website may not be accessed by web robots.

There is an important number of REL standards. Barlas (2006, p. 39) concludes that there are two main types: those developed as international standards (ISO REL is given as the example), and those developed more "akin to the open source development context", (ODRL is given as the example).

For general web services, the Web Services Policy (WS-Policy)[6] is a machine-readable language defined by the World Wide Web Consortium (W3C). There are several protocol specifications for different policies: security, reliable messaging etc., so its scope is far beyond rights expressions. It is designed to be used with W3C web services technologies, specifically WSDL (Web Services Description Language) and UDDI (Universal Description, Discovery and Integration), not with OGC web services.

In the geospatial information context, the OGC Geospatial Digital Rights Management Reference Model (GeoDRM RM) provides a conceptual, i.e. abstract, model for digital rights management of geospatial resources (Vowles, 2006). This reference model takes into consideration existing standards for the licensing of digital content and adapts them to the spatial domain, en-

[5] `http://www.robotstxt.org/orig.html`

[6] `http://www.w3.org/TR/ws-policy/`

visioning the creation of a "large-scale, open market in geospatial resources" (p. 13).

The ISO 19149:2011, Geographic information - Rights expression language for geographic information - GeoREL, defines an XML-based language to express rights for geographic information in order to allow for the creation of digital licenses for geographic information and services (ISO, 2011). It extends the ISO/IEC 21000-5:2004, Rights Expression Language (ISO REL) (ISO/IEC, 2004), and also takes from that standard the mechanisms to enforce and preserve rigths. The other foundational source of the ISO GeoREL is the ISO/DIS 19153 standard, currently "under development", which is based on the OGC GeoDRM Reference Model described in the previous paragraph.

The Open Digital Rights Language (ODRL) 2.0[7] is an open standard for policy expressions being developed in a W3C Community Group. It includes a well-defined XML encoding and a common vocabulary appropriate for general digital media. Its core model supports rights expressions for both commercial transactions and open access distributed content. ODRL can be embedded in a full DRM framework, but it does not provide one or make any requirements on how this framework should work.

Although it is generic, ODRL has been used in the geospatial domain. A geospatial profile was addressed, though not finished, for ODRL 1.1[8]. Bishr et al. (2007) provide an overview of different RELs in geospatial rights management and point out ODRL as their choice for implementation. Gabillon and Capolsini (2008) describe an ODRL profile specific for Web Map Services, also based on ODRL 1.1, fine-grained (e.g. it supports defining areas which can't be zoomed) and oriented towards DRM enforcement. Murti and Tadimeti (2011) have used ODRL 1.1 in a very simple DRM system for Spatial Data Infrastructures.

The existence of several DRM and REL standards which can be applicable to geospatial services, and the fact that research is needed to test their applicability to different scenarios and users, is noticed by the INSPIRE Network Services Drafting Team when they point out that although the access to a spatial data service may be restricted or subject to a license, there are not mature enough standards (Network Services Drafting Team, 2013, p. 23),

[7] `http://www.w3.org/community/odrl/`

[8] `http://odrl.net/Profiles/Geospatial/`

while The UK Location Architecture Interoperability Board - Business Interoperability Working Group (2012, Annex A) shows the existence of a number of different, incompatible implementations of DRM in the European location community. Longhorn and Gartmann (2012) also point out that the ISO 19149:2011 and ISO 19153 standards have been welcome in the geospatial field, but they are "a long way from being widely implemented" (p. 10).

3. A protocol for OGC web service cache policies

OGC web services allow for expressing some information about use "fees" and "access constraints" in their capabilities. However, neither the meaning nor the syntax of these elements is defined, as they are free text fields. This section describes a protocol to declare ISO/OGC service policies in a machine-readable format. This protocol is designed to regulate the behavior of harvesters which access data and metadata-providing OGC web services in order to download and cache those data and metadata.

The protocol consists of four components: a mechanism to express cache policies in ODRL 2.0, metadata to identify the protocol version expressed in the Dublin Core metadata standard (ISO, 2009), a solution to embed the cache policies in OGC web service capabilities and an algorithm that cooperative harvesters can implement to follow the cache policies established for those services. The cache policies and the protocol metadata are based on XML templates; if the cache policies of an OGC web service are described following one of these templates, a cooperative harvester can respect and follow those policies. The components of the protocol are addressed in the following sections.

3.1. ODRL 2.0 main elements

The ODRL 2.0 core model[9] central entity is the *Policy*, which may refer to *Permissions* and *Prohibitions*. Permissions allow *Actions* to be executed on *Assets*, e.g. the permission to download a certain file. Assets identify the contents subject to the Policies (e.g. a data file). *Constraints* allow to specify Permissions more precisely. *Parties* grant Permissions (*Role assigner*) or are granted Permissions (*Role assignee*). A Permission may be linked to *Duties*, which state that certain Actions may be executed by an assignee for the

[9]http://www.w3.org/community/odrl/two/model/

Permission to be valid (e.g. make a payment for download to be allowed). Prohibitions are used like Permissions, but they forbid Actions and can't be linked to Duties. The ODRL 2.0 core model is shown in UML in Figure 1.

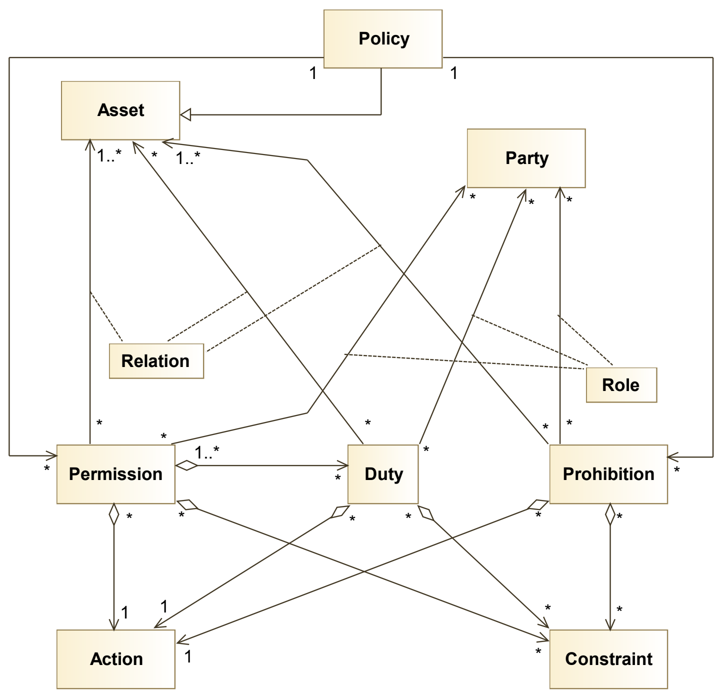

Figure 1: ODRL 2.0 core model in UML

The ODRL 2.0 common vocabulary[10] specifies the terms used by the core model and also includes additional semantics to express Policy Types, Permissions, Prohibitions, Constraints, Party Roles and Duties over Assets. Examples of the defined Policy Types are *agreements* or *offers*, and examples of Actions (for Permissions, Prohibitions and Duties) are *archive*, *delete* or *distribute*. Constraints have a name, operator and a rightOperand. Constraint names can be for instance *dateTime*, *language* or *version*, and the operators and rightOperands allow for fully specifying the constraints (e.g. "version eq 1.0" means that the version of an Asset must be exactly 1.0 for the constrained Permission to be issued).

[10] `http://www.w3.org/community/odrl/two/vocab/`

3.2. Cache policy templates

This section proposes three templates for expressing cache policies for OGC web services in ODRL 2.0. XML versions of these templates, compliant with the ODRL 2.0 encoding, are also provided in order to make it easy to tailor them to real OGC web services. The templates must take into consideration the different ODRL elements:

- Type of policy, *Set*: ODRL Set policies can be used as "instant licenses" that do not require explicit assigners or assignees.

- The main involved parties: The assigner is the owner of the service, which is already identified in the service capabilities, and the assignee is any harvester accessing the service, so we do not require parties to be made explicit in the policy.

- Assets with unique identifiers: We will be using Uniform Resource Identifiers (URIs) for the unique identifiers. We will have both whole service contents and distinct parts of these services contents (e.g. map layers) as assets.

- The allowed actions for the permissions (if any) and the forbidden actions for the prohibitions (if any): In the ODRL 2.0 Common Vocabulary, the action whose semantics are closer to the basic permission to access the contents of a data service is *read* ("The act of obtaining data from the asset") where some constraints can be applied if necessary. The action whose semantics are closer to cache is *archive* ("The act of persistently store the asset") so we need to express the permission to archive and the prohibition to archive. If there were any contradictions (e.g. permission and prohibition for the same action on the same asset), the ODRL file would not be correct and thus its meaning would be undefined.

- Constraints on the allowed actions (if any): Many ODRL constraints could be applicable, but we have chosen two because of their relevance, simplicity and the possibility to be followed by an automatic service harvester:

  - *ElapsedTime* ("A period of time in which the policy action can be exercised") as a constraint on the permission to archive: this allows to limit the amount of time that the cached data can be kept stored. It may be used when the service contents change often, so the service owner prefers caches to be updated within a certain period, or to indicate that a web service policies may change, so the permission given to archive is just temporary.

  - *Count* ("The number of times the corresponding entity may be exercised") in a given *timeInterval* ("recurring period of time in which the usage may be exercised") on the permission to read: this allows to limit the amount of requests that can be launched to a service in a given amount of time. Service owners that allow caching but are worried about the impact on performance of automatic harvesters can use these constraints.

  The rationale on these choices and possible future work on this issue are included in sections 5 and 6.

- Duties that must be fulfilled for the permissions to be granted (if any): As with constraints, service owners may think of a variety of duties to be imposed for the permissions to be valid, but most of them can't be followed by an automatic crawler. However, there are two actions which, as duties, are generic enough to cover many cases: *obtainConsent* ("The act of requiring explicit consent from a party to perform the action on the asset") and *reviewPolicy* ("The act of performing a manual review of the terms associated with the asset"). With these duties a service owner can express a variety of requirements in a text license, or even negotiate with every potential archiver. Of course, these duties can't be automatically fulfilled by an automatic harvester, but an automatic harvester can be programmed to require the explicit confirmation that these duties have been fulfilled before caching the services.[11]

The templates also include a section with some metadata about the protocol itself. Currently this section includes only a title and an identifier of the version, which is a simple addition that will make it easier to evolve the protocol in the future. As ODRL does not support this kind of information, we have chosen a simple implementation based on the Dublin Core

[11] Rationale on this issue is addressed in Section 5

metadata standard. Our implementation in the templates is fully compatible with ODRL 2.0, and ODRL 2.0 parsers which are not aware of its existence will ignore it.

Template 1: Permission to cache the contents of a service. The contents offered through this service may be cached, as long as the optional constraints are satisfied and the optional duties are fulfilled. Valid for: WMS (creating tiles), WMTS (downloading map tiles), WFS (downloading features), WCS (downloading coverages) and CSW (downloading metadata):

Elements	**Required and optional values**
Assets	The web service (required)
Permissions	*Read* (required) and *archive* (required)
Constraints	*ElapsedTime* (optional, on the permission to *archive*). *Count* in a given *timeInterval* (optional, on the permission to *read*)
Duties	*ObtainConsent* (optional, on any permission). *ReviewPolicy* (optional, on any permission)

A valid ODRL 2.0 XML encoding of this template is available at `https://gist.github.com/anonymous/3ddf2fb4fad57e0c1ff6`.

Template 2: Prohibition to cache the contents of a service. The contents offered through this service must not be cached. Although we could assume that without an explicit permission, prohibition must be understood, some potential users may request the possibility to be explicit about this prohibition. Valid for: WMS (creating tiles), WMTS (downloading map tiles), WFS (downloading features), WCS (downloading coverages) and CSW (downloading metadata):

Elements	**Required and optional values**
Assets	The web service (required)
Permissions	*Read* (required)
Prohibitions	*Archive* (required)

We are not considering any duty or constraint on the read permission, because caching is prohibited so these duties or constraints would fall out of the scope of a harvester.

A valid ODRL 2.0 XML encoding of this template is available at `https://gist.github.com/anonymous/76293e5b0dba65758ad0`.

Template 3: Permission to cache a subset of the contents of a service, and prohibition to cache another subset. A subset of the contents offered through this service may be cached, as long as the optional constraints are satisfied and the optional duties are fulfilled. This subset is formed by explicitly identified assets. Another subset of the contents offered trough this service (which must be disjoint with the first subset) must not be cached. This second subset is formed also by explicitly identified assets. Valid for: WMS layers (creating tiles), WMTS layers (downloading map tiles), WFS feature types (downloading features) and WCS coverages (downloading coverages). We have not considered it useful to allow for the download of a subset of the metadata offered by CSW services, as there is not a natural way to express these subsets:

Elements	**Required and optional values**
Assets	The web service and the layers, feature types, or coverages that can be cached (required)
Permissions	*Read* (on the web service; required). *Archive* (on the layers, feature types or coverage types; required if there are not any prohibitions)
Prohibitions	*Archive* (on the layers, feature types or coverage types; required if there are not any permissions)
Constraints	*ElapsedTime* (optional, on the permission to *archive*), *count* in a given *timeInterval* (optional, on the permission to *read*)
Duties	*ObtainConsent* (optional, on any permission), *reviewPolicy* (optional, on any permission)

Even with disjoint subsets, there are different combinations of permissions and prohibitions. The rule to be applied by the harvesters is as follows: Contents which are not explicitly prohibited can be cached. If there is an explicit permission, some constraints or duties may apply. This means that if a service owner wants to prevent some contents from being cached, a prohibition must be made explicit.

> A valid ODRL 2.0 XML encoding of this template is available at `https://gist.github.com/anonymous/ac2c8794ff0d53b4c387`.

3.3. Embedding cache policies in OGC web service capabilities

OGC web services have well-defined an easily accessible service capabilities (i.e. service metadata). The inclusion of use policies in these capabilities is addressed by means of the optional `<AccessConstraints>` element, which allows for including them as free text (the element `<Fees>` is also related). In order to embed the machine-readable cache policies in ODRL described in the previous section in an OGC web service, we propose to use the optional `<AccessConstraints>` elements in the service capabilities:

- This field will be filled in with any license-related issues in natural language, so that people can understand them.
- Any number of URLs may be included in that field (it is free text after all). The first of these URLs which after dereferencing provides a valid ODRL 2.0 XML document will be used.
- If no valid ODRL 2.0 XML document is found, or if this document does not match the templates in the protocol, the service is not covered by the protocol specified in this work.
- The mechanism to add this information to the capabilities of the service is implementation-dependent. As the information is filled in a field described by the ISO/OGC standards, any standard server must offer this kind of mechanism.

The significant fragment of a valid service capabilites document following the previous guidelines, could be as the following example shows:

```
<ows:ServiceIdentification>
  ... Other Elements in this section of the capabilities...
```

```
  <ows:AccessConstraints> You have free access to this service
  ... There is a machine-readable, ODRL 2.0 license in this URL:
  http://webserver.company.com/odrl_license.xml
  You should also check our general policies for data sharing in
  http://webserver.company.com/datapolicies
  </ows:AccessConstraints>
</ows:ServiceIdentification>
```

ODRL Assets are those digital elements subject to the given policies, and they must have unique identifiers. We are proposing two types of assets: the contents served trough OGC web services and the “natural” subsets of those contents (Layers, Feature Types and Coverages). For every kind of asset, we propose a way to build a unique identifier as a URI (URLs are URIs):

Asset	**Unique identifier**
Service contents	`ServiceURL`, where ServiceURL is a URL that gives access to the service capabilities
WMS Layer	`ServiceURL#LayerName`, where LayerName is the mandatory `<Name>` defined for the `<Layer>` element in the service capabilities
WMTS Layer	`ServiceURL#LayerIdentifier`, where LayerIdentifier is the mandatory `<Identifier>` in the `<Layer>` in the `<Contents>` element in the service capabilities
WFS Feature Type	`ServiceURL#FeatureTypeName`, where FeatureTypeName is the mandatory `<Name>` in the `<FeatureType>` element in the service capabilities
WCS Coverage	`ServiceURL#CoverageId`, where CoverageId is the mandatory `<CoverageId>` in the `<CoverageSummary>` in the `<Contents>` element in the service capabilities

An ISO/OGC service may have different URLs that provide access to it (e.g. aliases) or may have different URLs for its different operations (this is allowed by the standards). As we need to choose a unique identifier for the assets, we make it a requirement to provide a URL that actually gives access to the service capabilities. Using that URL as a prefix in the identifiers of the layers, feature types and coverages we make sure that those identifiers are unique[12]

3.4. Harvester algorithm

Harvesters access OGC web services, download (i.e. harvest) their contents and store (i.e. cache) them. Before proceeding to the download, a harvester designed to follow the cache policies established in section 3.2 must extract and process those policies. This is done in order to determine whether the harvesting is allowed and whether the constraints and duties can be satisfied.

Algorithm 1 checks the optional duties (reviewPolicy and obtainConsent) and makes sure that they can be fulfilled by requesting the explicit confirmation of a human. This confirmation may be implemented in different ways, for instance interactively before launching a harvesting session. A harvester may also check if these duties have changed since the last visit to that service, and request human intervention only if they have. The algorithm takes lists of services and contents (e.g. layers) that we want to cache. For each service, it obtains its ODRL read permission and duties (lines 4-6). If the duties associated to the read permission can't be fulfilled then the service will not be cached (lines 7-8). In another case, if it is harvesting the whole service, it obtains its ODRL archive permission and duties (lines 1-12), and if they can't be fulfilled that service will not be cached (lines 13-14). If only some contents of the service are being cached, it obtains the ODRL archive permissions and duties for those contents (lines 18-19). If they can't be fulfilled, then those contents will not be cached. It finally returns those services and contents which duties have been fulfilled (line 27).

Algorithm 2, which is designed to be executed after algorithm 1, checks that the permissions exist to cache the services and contents, makes sure that the constraints can be respected and proceeds to the caching. The algorithm takes lists of services and contents (e.g. layers) that we want to cache and whose duties have already been fulfilled by algorithm 1. First of all, it creates empty lists of services and contents to cache (lines 1-2). Then, for each service, if service caching is not explicitly prohibited and we want to harvest the whole service and service caching is permitted, it adds this service to the

[12]For example, the layer "Rivers", which must be a unique name inside my service capabilities, will be identified with the URI `http://myserviceurl.com#Rivers`. As long as myserviceurl.com gives access to the service capabilities, as required, that URI should be unique.

list of services to cache along with the constraints if they exist (they can be request rate, i.e. count per timeInterval, and elapsedTime) (lines 4-10). If service caching is not prohibited and we only want to cache some contents in the service, then it takes those contents whose caching is permitted and adds them to the list of contents to cache along with their constraints (lines 14-16) and also takes those contents whose caching is not prohibited and adds them to the list of contents to cache without any constraints (constraints must be attached to permissions) (lines 18-19). It finally caches the services and contents in the lists of services and contents to cache, following the constraints included along with them in those lists (line 26).

An experimental implementation of these algorithms in Python has been developed to test them, and is available at `https://github.com/rbejar/odrl-ogc-cache-policies`. Instructions to download and run the code are included. The implementation includes five tests that cover different aspects of the protocol while accessing three WFS services with different ODRL licenses (each one based on one of the provided XML templates). Information about the set-up of these services is also included.

4. Application to the EuroGeoSource information system

The EuroGeoSource project,[13] co-funded by the Competitiveness and Innovation Framework Programme (CIP), under the Policy Support Programme (PSP), Geographic Information Theme, of the European Union, has developed an Internet pilot information system[14] which provides access to geographical information on geo-energy and mineral resources on ten European countries.

The system has been developed following an SDI architecture based on INSPIRE principles and ISO/OGC standards (Béjar et al., 2013). The data-providing partners, most of them geological surveys, have developed a common data model, and they currently host WFS services that provide access to their data in that model. The EuroGeoSource web portal harvests periodically those WFS services, and stores the data in a central database in order to provide a good performance in data queries.

Several of these data-providing partners expressed concerns related to the harvesting of their WFS services, and to the degree of control that they

[13] `http://www.eurogeosource.eu/`
[14] `http://maps.eurogeosource.eu/`

Algorithm 1 Fulfill duties requiring human intervention

Require: *allServices*: services that we want to cache; *allContents*: service contents (e.g. layers) that we want to cache

Ensure: If the protocol version is understood by this harvester: *servicesOut*: services in *allServices* that have had their duties fulfilled; *contentsOut*: service contents in *allContents* that have had their duties fulfilled;

```
servicesOut ← allServices
contentsOut ← allContents
for service in allServices do
  serviceODRL ← getODRL(service)
  protocolVersion ← getProtocolVersion(serviceODRL)
  if protocolVersion is not understood by this harvester then
    return protocol version not supported
  end if
  serviceReadPerm ← getServiceReadPerm(serviceODRL)
  serviceReadDuties ← getServiceReadDuties(serviceReadPerm)
  if serviceReadDuties can't be fulfilled then
    Remove service from servicesOut
  else
    if the whole service is being harvested then
      serviceArchPerm ← getServiceArchPerm(serviceODRL)
      serviceArchDut ← getServiceArchDuties(serviceArchPerm)
      if serviceArchDut can't be fulfilled then
        Remove service from servicesOut
      end if
    else
      for c in all service contents included in allContents do
        contArchPerm ← getContentArchPerm(serviceODRL, c)
        contArchDut ← getContentArchDuties(contArchPerm)
        if contArchDut can't be fulfilled then
          Remove c from contentsOut
        end if
      end for
    end if
  end if
end for
return servicesOut, contentsOut
```

Algorithm 2 Cache services and contents whose duties have been fulfilled

Require: *allServices*: services that we want to cache; *allContents*: service contents (e.g. layers) that we want to cache; Both have had their duties fulfilled (they have been produced by Algorithm 1)

Ensure: If the protocol version is understood by this harvester, those services in *allServices* and contents in *allContents* that we can cache according to the protocol will have been cached, with the limits imposed by their duties.

```
servicesToCache ← [ ]
contentsToCache ← [ ]
for service in all services being harvested do
    serviceODRL ← getODRL(service)
    protocolVersion ← getProtocolVersion(serviceODRL)
    if protocolVersion is not understood by this harvester then
        return  protocol version not supported
    end if
    serviceReadPerm ← getServiceReadPerm(serviceODRL)
    if service caching is not prohibited in serviceODRL then
        if the whole service is being harvested then
            if service caching is permitted in serviceODRL then
                serviceArchPerm ← getServArchPerm(serviceODRL)
                Add (service, getRequestRate(serviceReadPerm), getElapsedTime(serviceArchPerm)) To servicesToCache
            end if
        else
            for c in all service contents included in allContents do
                if c caching is permitted in serviceODRL then
                    contentArchPerm ← getContArchPerm(serviceODRL, c)
                    Add (service#c, getRequestRate(serviceReadPerm), getElapsedTime(contentArchPerm)) To contentsToCache
                else
                    if c caching is not prohibited in serviceODRL then
                        Add (service#c, getRequestRate(serviceReadPerm),) To contentsToCache
                    end if
                end if
            end for
        end if
    end if
end for
Cache servicesToCache and contentsToCache with the limits of the request rates and keeping the date for those where a limit on the cache time is required
```

have on this. After interviewing those partners to understand their needs, we structured them as functional requirements. The system had to:

1. Allow them to express their permission to cache the contents of their WFSs if they chose to do so.
2. Allow them to express their prohibition to cache the contents of their WFSs if they chose to do so.
3. Erase all cached contents when they change the permission to a prohibition, or when they establish a prohibition when nothing had been explicitly expressed before.
4. Consider the lack of an explicit prohibition as an implicit permission. This had to be so because not every partner needed to express this kind of permissions in their services.

After some discussion with all the partners involved in the development of the system, we also established some non-functional requirements:

1. The system had to be as automated as possible (e.g. the service owners should not need to contact a human when they decided not to allow anymore the caching of their contents).
2. The solution had to be based on open standards, as the rest of the system.

The protocol for OGC web service cache policies proposed in this paper is based on the solution designed and implemented for the EuroGeoSource project, though it has been generalized to make it applicable to other services and use cases.

4.1. Cache policies and the EuroGeoSource WFS harvester

The EuroGeoSource data-providing partners are in full control of what is offered through their WFS services, and they can include natural language licenses in those services if they need to express use terms. However, as the web portal database harvester is a periodic, automatic process, several of these partners expressed an interest in having a way to have more control about this process. To address this issue, two cache policies have been defined:

- Explicit permission to cache the contents of a WFS service. This policy is similar to the one described in section 3.2, but without any duties or constraints. This supports functional requirement 1.

- Explicit prohibition to cache the contents of a WFS service. This policy is similar to the one described in section 3.2. This supports functional requirement 2.

These cache policies are expressed in ODRL 2.0 (this supports the non-functional requirement 2). The only permission / prohibition considered was *archive*, and the only asset was the WFS service, uniquely identified by its URL. The ODRL 2.0 XML document must be embedded in the `<AccessConstraints>` field of the WFS capabilities, as described in section 3.3.

The WFS harvester in the EuroGeoSource project has been developed with a free Spatial ETL (Extraction, Transformation and Load) tool called GeoKettle[15] (Waardenburg and Kerkenaar, 2012; Béjar et al., 2013). The harvester goes to the data providing partner WFSs and, unless there is an explicit prohibition as described above (functional requirement 4 says that without an explicit prohibition, implicit permission must be understood), it downloads the data, transforms it to the geodatabase model used and updates the database. This supports the non-functional requirement 1. If there is an explicit prohibition, and there are cached data, it erases them. This supports functional requirement 3.

The process is launched periodically by means of the task scheduler in the web portal server. The harvester is designed to process data in the EuroGeoSource model downloaded from the EuroGeoSource WFSs. A more generic, and reusable, harvester would be far simpler, as it would just need to download the data from a WFS, leaving any further transformation or processing to other tools.

5. Rationale and discussion

During this work, a number of decisions have been taken. The rationale behind them is discussed in this section. These decisions have been guided by our interest to provide a novel solution to a real problem, which is both easy to adopt and extensible if needed.

An acceptable balance between simplicity (both for the service owners and for the service harvesters) and functionality has been difficult to achieve. The first issue was the granularity of the assets. From the whole service, to

[15] http://www.spatialytics.org/projects/geokettle/

individual features in a WFS or bounding boxes in a WMS, there are many possibilities. Having considered both whole services and layers, feature types and coverages, we have covered the most common cases related to caching contents, though some extensions could be considered in future work (see section 6).

The other major issue related to simplicity includes constraints and duties. Without them, the degree of control offered to the service owners is pretty low (i.e. full prohibition or full permission). With too many of them, the number of possible combinations of rights and constraints may explode, making it very difficult for harvesters to respect all of them. We have chosen two constraints (elapsedTime and count) related to the needs expressed by the service owners in the EuroGeoSource project and by other examples like the Spanish Cadastre WMS, mentioned before, or the OpenStreetMap tile usage policy[16].

With regards to duties, we have chosen two, obtainConsent and reviewPolicy, which are very flexible because they allow to express anything as a free text license and require the harvester owners to read and accept them before proceeding. This goes beyond click-trough licenses in two ways: on the one hand, it makes it possible for an automatic harvester to "know" that there are duties that a human is required to fulfill. On the other hand, when these duties are not needed, the protocol is fully automatic. We think that this solution provides a reasonable balance, but more experiments could help to verify this point.

Besides choosing a small number of constraints and duties we have provided templates in order to make it easier for the OGC web service right-holders to use them (they just need to choose a template and "fill in" the blanks), and easier for the harvesters to support them (there is a limited number of combinations that they must understand).

We are not requiring parties to be made explicit in the ODRL licenses because that lowers barriers to adoption. As parties must be uniquely identified, that raises non-trivial questions (e.g. Would we need some kind of registry for OGC web service right-holders?). The kind of "instant license" that is created in this way fits our problem, and is perfectly right in ODRL.

[16]See `http://wiki.openstreetmap.org/wiki/Tile_usage_policy`. For instance, it expresses, in natural language, the prohibition of a "heavy use" of the map tile server without prior permission.

We could have chosen other options to link the ODRL cache policies with the service capabilities. One option would have been requiring a service catalog, and including the cache policy in the service metadata in that catalog. This would have been a problematic requirement as many OGC web services are not described in catalogs, or are described in catalogs which are not in control of the service owners. Other option is the "extended capabilities" in OGC/ISO services, that can be defined with any contents. This requires defining an XML schema and it makes it more difficult for the service owners to implement the solution, as they would need to change not only the contents but also the structure of their service capabilities.

5.1. An alternative implementation: GeoREL

Although ODRL 2.0 has proven itself as an adequate tool for this work, the ISO 19149:2011 (GeoREL) standard could have been used too. A detailed comparison of ODRL 2.0 and GeoREL is out of the scope of this paper, but this section highlights a number of differences between those standards.

GeoREL is intended to be part of the same DRM security system proposed by the ISO/IEC 21000 REL (ISO, 2011, p. 6). This is in contrast with the strict separation from REL and DRM that ODRL 2.0 proposes and that has been recommended to facilitate the adoption of DRM (Jamkhedkar et al., 2006). Taking DRM security into consideration has additional requirements. For instance, a GeoREL license (which is a complete rights expression) is valid if, and only if, it carries a valid digital signature of the resource owners or their recognized agents. Another example: the abstract test cases included in the GeoREL standard have two parts: license conformance and enforcement conformance.

GeoREL distinguishes between *GeoResources* (e.g geodata) and *GeoProcesses* (e.g. OGC web services) (resources are similar to assets in ODRL 2.0). ODRL 2.0 can work with any kind of asset, so it does not make any constraint on what an asset can be: it just requires assets to be uniquely identified. We would have used GeoProcess in a GeoREL implementation of our proposal.

GeoREL is specifically a geospatial standard, so it provides a number of geospatial concepts. For instance, GeoREL *GeoRights* (similar to the permitted actions in ODRL 2.0) provide, "out of the box", the means to express the right to make a *spatial transform* to a GeoResource. This happens too with *conditions* (similar to the constraints in ODRL 2.0) where, for instance, you have *spatial temporal* conditions already defined. We have not required these kind of geospatial concepts for our work, but in the future we could be needing some of them. We hope that if an ODRL 2.0 geospatial profile is developed, it does not "reinvent the wheel" with concepts that are already defined in GeoREL.

ODRL 2.0 distinguishes between *read* ("obtaining data from"), *copy* ("make and exact reproduction") and *archive* ("persistently storing"). In our proposal we have used read and archive. In GeoREL this distinction does not exist as such, so we would have chosen *extract/copy* ("extract a subset as a local copy, or make a full copy") to cover our needs. There is nothing specific to "caching spatial contents" neither in ODRL 2.0 nor in GeoREL.

With regards to the *issuers* (similar to the parties with role assigner in ODRL 2.0), they are required in GeoREL licenses, but not in ODRL 2.0, as it is mandatory to include a valid digital signature of the resource owners or their recognized agents. *Principals* (similar to the parties with role assignee in ODRL 2.0) are mandatory in GeoREL too, but it is not difficult to create a "public user" (ISO, 2011, Example 4)), which would haven been the best choice for our needs. The mandatory digital signature for the resource owners could have been identified as a barrier to adoption in our proposal. One option for an alternative implementation in GeoREL could have been not requiring a GeoREL license and accepting instead a GeoREL *grant*, which is a fragment of a complete license.

6. Conclusions and future work

This paper proposes a protocol to specify cache policies for OGC web services in ODRL 2.0 that can be followed by cooperative harvesters. We have used ODRL 2.0 because our specific interest of using a REL not related to any particular DRM system. This separation is a recommended practice even when the final objective is the adoption of a DRM framework (Jamkhedkar et al., 2006) or the use of access control technologies (The UK Location Architecture Interoperability Board - Business Interoperability Working Group, 2012, p. 4), although DRM is not a solution for everybody.[17] This protocol addresses a current necessity, but it can also be a stepping stone towards DRM for those interested.

[17]Even from an economic point of view, DRM may not always be the best option (Ahn and Shin, 2012). Other authors have suggested that a careful balance between enforcement technologies and trust needs to be achieved (Barlas, 2006).

The kind of services considered are useful for the environmental community and there are many of them available. Only in Europe, in 2010 there were found 3712 WMSs, 1492 WFSs, 1156 WCSs and 45 CSWs (Lopez-Pellicer et al., 2011, Table 3). The protocol can be applied to other kinds of services by defining a way to embed the ODRL policy files in those services metadata, and a way to identify those services assets (e.g. their layers, feature types or equivalent).

A preliminary version of this protocol was defined and tested during the EuroGeoSource project. That project was a good example of the big number of legal and technical circumstances of the different data and service providers. This suggested that a solution should be able to cover substantially different cases and still be simple enough to facilitate its adoption. The protocol has been defined taking into consideration the core needs of the EuroGeoSource project, and extending it to cover other expected situations while trying to keep low barriers for its adoption (i.e. keeping it simple). However, there are a number of issues that could be considered as extensions in future versions:

- The *resolution* constraint, included in the ODRL 2.0 common vocabulary, may be useful for some of the considered services (i.e. those offering raster contents). It could be used to define allowed ranges of resolutions for caching.

- The *watermark* duty, included in the ODRL 2.0 common vocabulary, can be used to force watermarking tiles cached from a WMS.

- Spatial constraints can be used to limit the areas that can be cached. Spatial areas defined with codes are supported by the common vocabulary of ODRL 2.0., but bounding boxes defined with coordinates are not. This could be included as one step towards a geospatial profile of ODRL 2.0.

- Service owners may be interested in banning the harvesters at the times of the day where they have more users, but allowing them for instance at night. There are several constraints in the common vocabulary of ODRL 2.0 related to time that can be used for this.

ODRL 2.0 supports the creation of profiles in order to address the needs of different thematic communities. Environmental and geospatial profiles of

ODRL 2.0 are long-term objectives that could contribute to an improvement in the current situation of the expression of rights for these data and services, and we consider that they could be addressed by the environmental research community.

ODRL 2.0 has been chosen to express the cache polices. The ISO 19149:2011 (GeoREL) would be the other major option in the environmental geospatial context of this work. There are also other possibilities, but none of these options are considered mature enough (Network Services Drafting Team, 2013; Longhorn and Gartmann, 2012). Research is thus needed both to test the different standards in different problem domains, and to test and compare these standards in use cases ranging from simple rights expressions to full DRM schemes. Besides this, a Web where geospatial and non-geospatial services interoperate will require that different RELs and DRM systems interoperate too. This opens other possibilities for future work, including the creation of similar caching protocols at other levels, from HTTP to general web services, and the development of interoperation mechanisms among them.

7. Acknowledgments

This work has been partially funded through the EuroGeoSource project (project number 250532), from the European Union's ICT Policy Support Programme as part of the Competitiveness and Innovation Framework Programme, by the Spanish Government (project TIN2012-37826-C02-01) and by the National Geographic Institute (IGN) of Spain. We also want to thank the members of the EuroGeoSource consortium their help, suggestions and hard work, and the valuable suggestions from the reviewers.

Ahn, I., Shin, I., 2012. On the optimal level of protection in DRM. Information Economics and Policy 22, 341–353.

Ames, D. P., Horsburgh, J. S., Cao, Y., Kadlec, J., Whiteaker, T., Valentine, D., 2012. HydroDesktop: Web services-based software for hydrologic data discovery, download, visualization, and analysis. Environmental Modelling & Software 37, 146–156.

Bach, K., Schäfer, D., Enke, N., Seeger, B., Gemeinholzer, B., Bendix, J., 2012. A comparative evaluation of technical solutions for long-term data repositories in integrative biodiversity research. Ecological Informatics 11, 16–24.

Barlas, C., 2006. Digital Rights Expression Languages (DRELs). JISC Technology and Standards Watch 6 (3), 1–42.

Behrens, L., Reichling, J., 2011. "GeoBusinessLicence" - one licence for all. In: Proceedings of the INSPIRE conference 2011, Edinburgh, Scotland (United Kingdom).
URL http://inspire.jrc.ec.europa.eu/events/conferences/inspire_2011/abstracts/9.pdf

Bishr, M., Wytzisk, A., Morales, J., 2007. GeoDRM: Towards digital management of intellectual property rights for Spatial Data Infrastructures. In: Onsrud, H. (Ed.), Research and Theory in Advancing Spatial Data Infrastructure Concepts. ESRI Press, Redlands, CA, USA., pp. 245–260.

Béjar, R., Fruijtier, S., Usón, M., March 2013. WP8 - system development and implementation. Public Deliverable D8.1, EuroGeoSource project consortium.
URL http://www.eurogeosource.eu/docs/D8.1._EGS_Report_System_Development.pdf

Béjar, R., Latre, M. Á., Nogueras-Iso, J., Muro-Medrano, P., Zarazaga-Soria, F. J., 2012. An RM-ODP enterprise view for spatial data infrastructures. Computer Standards & Interfaces 34, 263–272.

Côte, C. M., Moran, C. J., Hedemann, C. J., Koch, C., 2010. Systems modelling for effective mine water management. Environmental Modelling & Software 25 (12), 1664–1671.

Deng, Y., Wu, Q., 2010. Research on the harvest and cascade of catalogue service in GeoGlobe service platform. In: Geoinformatics, 2010 18th International Conference on.

Fegraus, E. H., Andelman, S., Jones, M. B., Schildhauer, M., 2005. Maximizing the value of ecological data with structured metadata: An introduction to ecological metadata language (EML) and principles for metadata creation. Bulletin of the Ecological Society of America 86 (3), 158 – 168.

Frehner, M., Brändli, M., 2006. Virtual database: Spatial analysis in a Web-based data management system for distributed ecological data. Environmental Modelling & Software 21 (11), 1544–1554.

Gabillon, A., Capolsini, P., October 2008. An ODRL Web Map Service profile. In: Proceedings of the 2008 joint Virtual Goods and ODRL Workshop. Poznan, Poland, pp. 23–43.

García, R., Verdú, E., Regueras, L. M., de Castro, J. P., Verdú, M. J., 2013. A neural network based intelligent system for tile prefetching in web map services. Expert Systems with Applications 40 (10), 4096–4105.

Gebhardt, S., Wehrmann, T., Klinger, V., Schettler, I., Huth, J., Kunzer, C., Dech, S., 2010. Improving data management and dissemination in Web based information systems by semantic enrichment of descriptive data aspects. Computers & Geosciences 36, 1362–1373.

González, V., García, I., del Moral, F., de Haro, S., Sánchez, J. A., Simón, M., 2011. Impact of unconfined sulphur-mine waste on a semi-arid environment (Almería, SE Spain). Journal of Environmental Management 92 (6), 1509–1519.

Granell, C., Díaz, L., Gould, M., 2010. Service-oriented applications for environmental models: Reusable geospatial services. Environmental Modelling & Software 25, 182–198.

Granell, C., Díaz, L., Schade, S., Ostländer, N., Huerta, J., 2013. Enhancing integrated environmental modelling by designing resource-oriented interfaces. Environmental Modelling & Software 39, 229–246.

ISO, 2005. Geographic information - Services. International standard; ISO 19119. International Organization for Standardization, Geneva, Switzerland.

ISO, 2009. Information and documentation - The Dublin Core metadata element set. International standard; ISO 15836. International Organization for Standardization, Geneva, Switzerland.

ISO, 2011. Geographic information - Rights expression language for geographic information - GeoREL, 1st Edition. International standard; ISO 19149. International Organization for Standardization, Geneva, Switzerland.

ISO/IEC, 2004. Information technology - Multimedia framework (MPEG-21) - Part 5: Rights Expression Language, 1st Edition. International standard;

ISO/IEC 21000-5. International Organization for Standardization / International Electrotechnical Commission, Geneva, Switzerland.

Jamkhedkar, P. A., Heileman, G. L., Martínez-Ortiz, I., 2006. The problem with rights expression languages. In: In DRM '06: Proceedings of the ACM workshop on Digital rights management, Alexandria, Virgina, USA. ACM Press, pp. 59–68.

Janßen, T., Schmidt, M., Dressler, S., Hahn, K., Hien, M., Konaté, S., Lykke, A. M., Mahamane, A., Sambou, B., Sinsin, B., Thiombiano, A., Wittig, R., Zizka, G., 2011. Addressing data property rights concerns and providing incentives for collaborative data pooling: the West African Vegetation Database approach. Journal of Vegetation Science 22 (4), 614–620.

Jones, O. R., Clutton-Brock, T., Coulson, T., Godfray, H. J., 2008. A web resource for the UK's long-term individual-based time-series (lits) data. Journal of Animal Ecology 77 (3), 612–615.

Laniak, G. F., Olchin, G., Goodall, J., Voinov, A., Hill, M., Glynn, P., Whelan, G., Geller, G., Quinn, N., Blind, M., Peckham, S., Reaney, S., Gaber, N., Kennedy, R., Hughes, A., 2013. Integrated environmental modeling: A vision and roadmap for the future. Environmental Modelling & Software 39, 3–23.

Latre, M. Á., Lopez-Pellicer, F. J., Nogueras-Iso, J., Béjar, R., Zarazaga-Soria, F. J., Muro-Medrano, P. R., 2013. Spatial Data Infrastructures for environmental e-government services: the case of water abstractions authorisations. Environmental Modelling & Software 48, 81–92.

Li, M., Zhu, X., Mei, Y., 2011. Incremental harvesting model of distributed geospatial data registry center based on CSW. In: Geoinformatics, 2011 19th International Conference on.

Lia, Z., Yangb, C. P., Wua, H., Lic, W., Miaoa, L., 2011. An optimized framework for seamlessly integrating OGC web services to support geospatial sciences. International Journal of Geographical Information Science 25 (4), 595–613.

Liu, H., Nie, Y., 2010. Tile-based map service GeoWebCache middleware. In: Intelligent Computing and Intelligent Systems (ICIS), 2010 IEEE International Conference on. pp. 692–697.

Longhorn, R., Gartmann, R., 2012. Digital rights management and licensing of geospatial data and services. In: Proceedings of the 13th global spatial data infrastructure world conference (GSDI 13), 14-17 May 2012, Québec City, Canada.

Lopez-Pellicer, F. J., Béjar, R., Florczyk, A. J., Muro-Medrano, P. R., Zarazaga-Soria, F. J., 2011. A review of the implementation of OGC web services across Europe. International Journal of SDI Research 6, 168–1867. URL http://ijsdir.jrc.ec.europa.eu/index.php/ijsdir/article/view/233/299

Lotz, T., Nieschulze, J., Bendix, J., Dobbermann, M., König-Ries, B., 2012. Diverse or uniform? — intercomparison of two major German project databases for interdisciplinary collaborative functional biodiversity research. Ecological Informatics 8, 10 – 19.

Michener, W. K., Porter, J., Servilla, M., Vanderbilt, K., 2011. Long term ecological research and information management. Ecological Informatics 6 (1), 13–24.

Murti, K. C. S., Tadimeti, V. R., 2011. A simplified GeoDRM model for SDI services. In: Proceedings of the 2011 International Conference on Communication, Computing & Security. pp. 545–548.

Network Services Drafting Team, March 2013. INSPIRE Spatial Data Services and services allowing spatial data services to be invoked– Draft Implementing Rules. Tech. rep.
URL http://inspire.jrc.ec.europa.eu/documents/Spatial_Data_Services/Draft_IR_SDS_and_Invoke_3.0.pdf

Peckham, S. D., Goodall, J. L., 2013. Driving plug-and-play models with data from web services: A demonstration of interoperability between CSDMS and cuahsi-his. Computers & Geosciences 53, 154–161.

Pla, M., Lleopart, A., 2010. Updating of vector databases at the Institut Cartogràfic de Catalunya. In: ISPRS Joint Workshop on: Core Spatial Databases - Updating, Maintenance and Services - from Theory to Practice. Haifa, Israel.

Salvati, P., Balducci, V., Bianchi, C., Guzzetti, F., Tonelli, G., 2009. A WebGIS for the dissemination of information on historical landslides and floods in Umbria, Italy. Geoinformatica 13, 305–322.

The UK Location Architecture Interoperability Board - Business Interoperability Working Group, 2012. Access control and rights management position statement v1.0. Tech. rep., UK Location Programme.
URL http://data.gov.uk/sites/default/files/Access%20Control%20and%20Rights%20Management%20PS%20v1%200.pdf

Vodden, P., Gartmann, R., 2011. Lessons learned implementing data and service licensing. In: Proceedings of the INSPIRE conference 2011, Edinburgh, Scotland (United Kingdom).
URL http://inspire.jrc.ec.europa.eu/events/conferences/inspire_2011/abstracts/72.doc

Vowles, G. (Ed.), December 2006. The OpenGIS Abstract Specification - Topic 18: Geospatial Digital Rigths Management Reference Model (GeoDRM RM). No. OGC 06-004r4. Open Geospatial Consortium Inc.

Waardenburg, F., Kerkenaar, E., March 2012. WP5 - interoperable geo-resources spatial data web services. Public Deliverable D5.1, EuroGeoSource project consortium.
URL http://www.eurogeosource.eu/docs/D5_1_EGSource_Service_Implementation.pdf