Where to Publish and Find Ontologies?
A Survey of Ontology Libraries

Mathieu d'Aquin,[a] Natalya F. Noy[b]

[a]*Knowledge Media Institute (KMi), The Open University, Milton Keynes, UK*

[b]*Stanford Center for Biomedical Informatics Research, Stanford University, Stanford, CA, USA*

Abstract

One of the key promises of the Semantic Web is its potential to enable and facilitate data interoperability. The ability of data providers and application developers to share and reuse ontologies is a critical component of this data interoperability: if different applications and data sources use the same set of well defined terms for describing their domain and data, it will be much easier for them to "talk" to one another. *Ontology libraries* are the systems that collect ontologies from different sources and facilitate the tasks of finding, exploring, and using these ontologies. Thus ontology libraries can serve as a link in enabling diverse users and applications to discover, evaluate, use, and publish ontologies. In this paper, we provide a survey of the growing—and surprisingly diverse—landscape of ontology libraries. We highlight how the varying scope and intended use of the libraries affects their features, content, and potential exploitation in applications. From reviewing eleven ontology libraries, we identify a core set of questions that ontology practitioners and users should consider in choosing an ontology library for finding ontologies or publishing their own. We also discuss the research challenges that emerge from this survey, for the developers of ontology libraries to address.

Key words: ontology, survey, ontology library

1 The Need for Ontology Libraries

Semantic Web standards enable data interoperability by creating a vast, distributed data space where users and software agents can publish and access information from many different sources. They can use, integrate, and aggregate this information, regardless of its provenance and physical location. The use of standard formats such as RDF guarantees interoperability at a *syntactic* level, making it possible for applications to reuse data and to link diverse data. Ontologies represent the essential technology that enables and facilitates interoperability at the *semantic* level, providing a formal conceptualization of

Email addresses: `m.daquin@open.ac.uk` (Mathieu d'Aquin,), `noy@stanford.edu` (Natalya F. Noy).

the data which can be shared, reused, and aligned. As the result, the current explosion in the amount of data published on the Semantic Web [25] is accompanied by a rapid increase in the number of ontologies that are being developed and made available online [31,36]. Naturally, it is more cost-effective for data providers to reuse existing, preferably well-established and well-tested ontologies to describe their data, rather than to develop ontologies from scratch. Furthermore, ontology reuse significantly facilitates data interoperability: if different data providers use the same ontology to describe their data, they can integrate the data much more easily.

In the last few years, we have witnessed several notable successes of ontology reuse. For example, the FOAF ontology [26] is commonly used to describe data in social networking, and the GoodRelations ontology [44] is quickly becoming the standard for describing products and offerings from commercial organisations. Communities of workers in specific domains have also reached agreement on foundational ontologies, achieving a high level of reuse for these specific ontologies. For example, many biomedical researchers use the Gene Ontology [41] to annotate their data.

However, these examples of reuse are still rather isolated. Indeed, in order for ontologies to perform their role—facilitating interoperability between different systems and datasets—users must be able to find relevant ontologies quickly and easily. They also must be able to share their ontologies in such a way that others can easily discover, access, and reuse them. Today, a user who needs to find an ontology that is appropriate for her application faces a number of significant challenges. First, she must find the ontologies that deal with her subject domain. Second, she must sift through the ontologies that she finds and determine which ones cover the domain sufficiently well for her to be able to use the ontology in her application, and she also must somehow evaluate the quality of the ontology to help her determine if the ontology will work for her application. Third, she often must be able to access the ontology in a specific format that works with her application (e.g., download an OWL file, perform a SPARQL query, use Web service calls to access ontology content, and so on). Finally, if she ends up extending an existing ontology or creating her own, she might want to share it with her colleagues or with the broader research community. The growing number of online **ontology libraries** are beginning to address these challenges, enabling users to find, reuse, and publish ontologies, and thus facilitating eventual data interoperability on the Semantic Web.

Researchers have used many different names to refer to systems for collecting ontologies and making them available: ontology directory, ontology repository, ontology library, ontology archive. We will use the term *ontology library* to refer to these systems. We broadly define an ontology library as *a Web-based system that provides access to an extensible collection of ontologies with the primary purpose of enabling users to find and use one or several ontologies from this collection*. While we explicitly mention "Web-based" as part of the definition, to the best of our knowledge, all ontology libraries that exist or existed in the past, are accessible through the Web. We also specify that ontology libraries should contain a *collection* of ontologies, to distinguish from ontology search engines such as Swoogle [36] or Watson [34], which automatically crawl the Web to index ontologies rather than collect them. Our definition discusses *extensible* collections and thus excludes the sites where there is no expectation of significantly broadening the collection with time (e.g., the set of SWEET ontologies [15] for environmental sciences). Our definition identifies the primary purpose of the collection as enabling users to find and use *ontologies*. Thus we exclude from our survey the collections that focus on data, such as the Linked Open Data collection of datasets [25].

Many researchers and developers have identified the availability of ontology libraries as a crucial component of the Semantic Web infrastructure [23]. Systems such as the Ontolingua server [39], the DAML ontology library [30], the Protégé ontology library [12], the Ontaria ontology directory [62], and SchemaWeb [14] pioneered the idea of ontology libraries and created the first ontology collections. Many of these early systems are no longer actively supported or maintained. However, a new generation of ontology libraries is replacing them. These libraries provide more advanced features and are able to handle the rapidly increasing number of ontologies that are becoming available.

In this paper, we provide a survey of this growing—and surprisingly diverse—landscape of ontology libraries. Our investigation shows that the libraries that exist today are created for different purposes and thus provide greatly varying functionalities and have very different scope. Therefore, a user who wants to reuse an ontology from the set of hundreds or thousands of ontologies available on the Web, must not only be able to use an ontology library to help her sort through this variety, but also must know *which* ontology library to use. We identify the common elements of the libraries and the different features that are affected by their design goals and scope. Depending on the user's requirements, she can determine which library would be the most appropriate one for her. We use our survey of eleven ontology libraries to identify a set of questions that *ontology practitioners and users* can ask in order to choose an appropriate ontology library to use. Our survey will also help *ontology developers* determine where they can publish their ontologies, how they can publish them, and which libraries they should use to expose their ontologies to other users in their communities. We believe that our survey will also be useful to the *developers and maintainers of the ontology-library systems*: it can serve as a reference, identifying reusable (positive and negative) experiences, as well as the remaining technological and research challenges related to the publication and reuse of ontologies.

We start this paper by discussing related work (Section 2). We briefly describe each system that we included in our survey in Section 3. Section 4 provides an overview of the features that we consider in our comparison. We discuss these features in detail in the context of the eleven ontology libraries in Sections 5 through 7. Specifically, Section 5 compares the purposes for which these systems were created and their coverage. In Section 6, we characterize the content of the libraries. We discuss the key functions that the libraries provide to their users in Section 7. After comparing all the systems, we discuss the implications for ontology practitioners and system developers in Section 8. We provide conclusions in Section 9. We include an Appendix with more details on various features of the ontology libraries.

2 Related Work

Ding and Fensel published one of the earliest surveys of ontology libraries in 2001 [37]. That survey focused on a small number of features and included nine libraries. At the time, the libraries supported a heterogeneous and often non-overlaping collection of ontology languages (W3C standards, such as RDFS and OWL, were not widely adopted yet). Interestingly, some of the themes that Ding and Fensel identified almost ten years ago are present in today's ontology libraries: support for ontology versioning is lacking in many of them; many do not support ontology editing (and it is still not clear whether or not they should). Searching and browsing has come a long way, as has the scale of the libraries in terms of the number of ontologies and their size.

A later survey [20] presented a different set of libraries, mostly focusing on the "server" functionalities of the libraries: what is the type of access that they provide and what stages in the ontology lifecycle they support. The survey did not focus on the type of content the libraries have or how the content is collected or maintained.

Many of the systems that these and other earlier surveys include, are no longer actively maintained. Indeed, almost none of the systems that we survey here are included in the earlier surveys. While many of the dimensions that these surveys included are important for comparing today's ontology libraries (and we use them in this paper), the newer systems bring up a new set of issues that the earlier surveys do not address. These issues include the need for ontology evaluation and ranking, and ontology metrics; the social mechanisms for creating and maintaining the content of the ontology libraries and the corresponding metadata; the need to create and maintain mappings between ontologies, as more and more overlapping ontologies become available; the infrastructure requirements to enable interoperability with other data on the Semantic Web; service-oriented architecture; scalability in terms of the sizes of the ontologies, etc. Furthermore, the advances in the Web-browser technology enabled a far richer set of user-interaction and visualization features than were possible several years ago. Many of the newer ontology libraries take full advantage of these developments.

In 2008, the Ontolog forum—a community of ontology practitioners—has launched the Open Ontology Repository Initiative (OOR) [23]. The members of the initiative work together on identifying desiderata and requirements for an ontology repository (*ontology library*, in our terms), with the main focus on the openness of content. The scope of OOR goes beyond ontologies, to include database schemas and XML schemas. A number of dimensions that we discuss in this survey were fleshed out during our discussions with the OOR members.

3 Ontology Libraries

We selected eleven ontology libraries from a new generation of ontology libraries that has emerged recently. As we will see later, these systems vary in many ways, including their size, purpose, the formats they handle, the features they provide, or the way they give access to the collected ontologies. Before analyzing these differences, we give a brief description of each of the systems that we included in this survey.

BioPortal [50] is a library of biomedical ontologies developed by the National Center for Biomedical Ontology. BioPortal focuses on community-based features for ontology annotation and evaluation, enabling users to provide comments and reviews on ontologies, contribute mappings between ontologies, and custom-tailored views on ontologies. BioPortal provides visualization of ontologies, search across all ontologies, and REST service access to the ontologies and their metadata. There are two other libraries that are built on the BioPortal code base and thus share most of the features of BioPortal: The Marine Metadata Initiative (MMI) is an e-science community supported by NSF that has adopted BioPortal as its ontology library. The MMI Ontology Registry ad Repository [5] extends the BioPortal code base with several functionalities, such as more extensive support for mappings, and access using query language SPARQL. The OOR community maintains another BioPortal clone [6] that is not limited to biomedical ontologies but is rather a generic ontology library.

Cupboard [33], created as part of the NeOn Project, is built around the idea of *spaces*, enabling each user to create her own ontology space, containing and relating the ontologies she has selected. Cupboard provides a number of functionalities in the context of these ontology spaces, supporting ontology developers in publishing and sharing their ontologies, ontology users in finding and reusing ontologies, and application developers in using ontologies in their applications. In particular, Cupboard provides users with the ability to search in dedicated ontology spaces and across the entire library, to rate and comment ontologies, and to relate ontologies in an ontology space through alignments. In addition, a plugin for the NeOn Toolkit ontology editor [35] allows ontology developers to reuse elements of shared ontologies directly within the ontology engineering environment.

The OBO Foundry initiative [58] aims at creating a set of well-documented and well-defined biomedical ontologies that are designed to work with one another. The OBO Foundry has an editorial process defining which ontologies become part of that collection. In order for an ontology to be included in the library as an OBO Foundry candidate, the editors must approve the ontology, verifying, among other things, that it is publicly available, has plurality of users, is documented properly, uses an appropriate identification scheme, that its content does not overlap with other ontologies, or the authors are committed to remove overlap. The OBO Foundry repository does not have a dedicated software platform; it uses a sourceforge repository to maintain ontologies and ontology versions.

oeGov [9] is a an initiative dedicated to the creation and collection of ontologies used for e-governement, run by TopQuadrant. oeGov relies on a blog system to publish ontologies, comment on them, and review them.

OLS [29], the Ontology Lookup Service, developed at the European Bioinformatics Institute, is a library of biomedical ontologies, with the main goal of providing query and browsing access to a set of ontologies used for annotating biomedical data. OLS contains ontologies in the OBO format [42]. It provides search features to find descriptions of entities in these ontologies. It also provides Web services access (through SOAP services) to the ontologies. A number of large biomedical projects use OLS as their main source of terms.

OntologyDesignPatterns.org (ODP) [10] is a portal for a catalogue of ontology design patterns. While it does not contain large domain ontologies per se, many of the ontology patterns in ODP are in fact ontologies themselves, or components that are integrated into other ontologies. For example, there are patterns for representing climactic zones or observations. Therefore, we include ODP as one of the ontology libraries in our survey as a *bona fide* ontology library. As part of its role of collecting and publishing ontology design patterns, ODP provides ontology library capabilities, including the provision of rich metadata and the automatic generation of documentation for ontology design patterns. The portal is supported by a quality committee, reviewing and approving candidate design patterns for inclusion to the catalogue.

OntoSelect [27] is an ontology library that provides advanced search mechanisms (by keywords and by topic), with ranking mechanisms based on a set of measures on the ontologies (e.g., number of imports, number of languages included). OntoSelect focuses on providing ontologies and related services that facilitate annotation and natural-language processing in multiple languages, supporting searches that are restricted to a particular language.

OntoSearch2 [60] focuses on providing efficient querying mechanisms across hundreds of ontologies. It implements these querying mechanisms as an extension of the SPARQL language, reducing the set of ontologies into the DL-Lite formalism to enable the use of inferences in real time, during the querying process. It also provides a simple keyword-based search mechanism over its collection of ontologies.

The ONKI ontology server [45] supports the Semantic Web infrastructure run by the Finnish government. The server contains the ontologies used to support several government-run services. Ontologies on the ONKI server cover areas of business, public administration, culture, geography, and also include several international thesauri. There are also upper ontologies, which define fundamental distinctions in the domain and many ontologies in the library link to them. Many ontologies are available in multiple languages (e.g., Swedish and Finnish). The ONKI server provides a web service and API access to its ontologies.

The TONES repository [18] is designed as a library of OWL ontologies that might be of use to tool developers for testing purposes. Thus, its maintainers collected ontologies that span the range of expressive power, particularly for the DL ontologies, vary in size, and exhibit different features. The interface provides browsing features based on OWLSight [11] and the ability to sort ontologies according to parameters that determine their complexity (e.g., the DL expressivity and the number of logical axioms).

Schema-Cache [13] developed by Talis, is a simple online system for searching and browsing a collection of RDF vocabularies, based on the Talis Platform. It provides keyword search and navigation mechanisms in a manually gathered collection of ontologies (schemas) and which can be accessed in a variety of formats.

Note that the list above is not exhaustive. We limited our selection to the libraries that (1) are active; (2) focus on providing access to ontologies; (3) are non-trivial in scope. Given that this field is developing very fast, it is possible that our survey misses some of the most recent developments. However, we hope that the comparison framework that we provide would enable readers to apply it to evaluating new ontology libraries that become available.

4 Characteristics of Ontology Libraries

Our survey of ontology libraries shows that the features of the libraries vary greatly. These features are often driven by the scope and purpose of the library, the specific user requirements that the library developers were addressing, and by the maturity of the software. In this section, we outline the features that we will use to compare and contrast the ontology libraries through the rest of the paper. Our main goal for the survey is to help users understand what are the different types of ontology libraries that exist today and which libraries are appropriate for which task. Hence, we selected the features that help us achieve that goal, by looking at the content and key functions of the libraries.

Figure 1 summarizes these features. We organize our feature review into three main themes:

Purpose and coverage: While the general goal of all ontology libraries is to provide access to a set of ontologies, the specific goals for this access, the coverage required by the users, and the size of the library vary greatly. Ontology libraries can be limited to a particular domain (e.g., biomedicine), type of ontologies (e.g., OWL ontologies), or be rather general. Also, the libraries are often designed with specific types of applications in mind, determining the type of access to the content that they provide (Section 5).

Content: A key feature of any ontology library is, naturally, its content (Section 6). We specifically look at the way the content is collected and selected

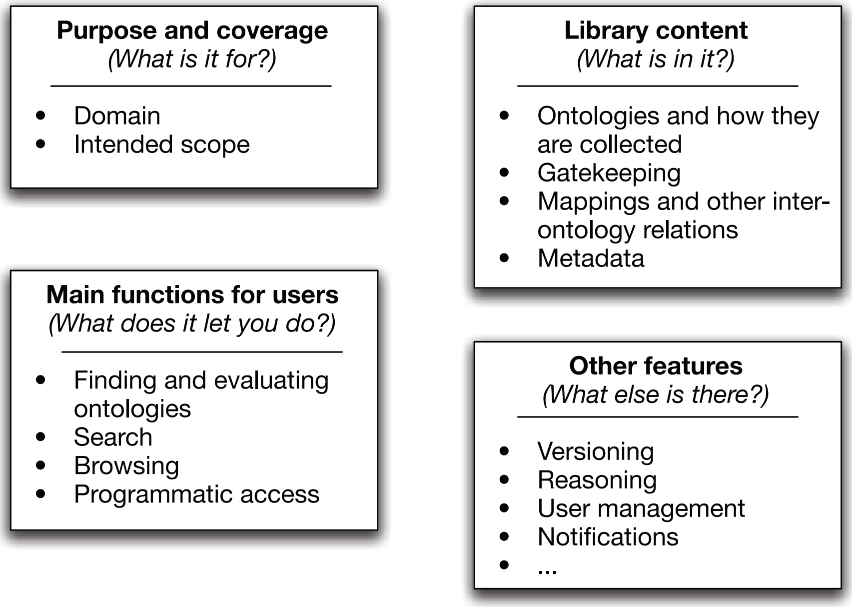

Fig. 1. The summary of the ontology-library features that we discuss in this survey. We discuss Purpose and coverage in Section 5, library content in Section 6, main functions in Section 7, and other features in the Appendix.

for each library, at specific characteristics of this content (size, format), and whether the library includes content other than the ontologies, such as ontology metadata and mappings.

Key functions: We consider the key functions that the library provides to its users, such as the ability to find and evaluate ontologies, facilities to search and browse, and to access ontologies and other library content programmatically (Section 7).

Our Appendix includes comparison of a variety of other features that may be of interest to those who develop similar tools or to those who would like to find out more details about the libraries (e.g., support for ontology versioning, user management, and technological infrastructure).

5 Purpose and Coverage

Through the rest of this paper, we detail the great variety of characteristics of various ontology libraries. Much of this variety comes from the different scope and purpose for which the libraries were designed. This initial set of characteristics addresses the questions of who are the target audience for the library, what is the user community that the library is addressing, what are the main use cases and tasks that the library can be used for. Specifically, we characterize the **purpose** and **domain coverage** of an ontology library (Table 1):

Purpose: What is the primary intended use of the ontology library? A library may be designed, for example, to assist users in finding an ontology of interest, to provide an easy access point for using ontologies in applications, to provide a social space for ontology developers, to provide a set of ontologies for testing reasoners, and so on. These design decisions greatly affect the content of the library (Section 6) and the set of functions that it provides (Section 7 and the Appendix). TONES, for example, aims to provide a broad set of ontologies that exercise all the possible features of the OWL language and their combination so that developers of inference engines can use the ontologies in the library for comprehensive tests of their reasoners. Thus, TONES focuses on the breadth of the OWL features rather than content and provides statistics, characteristics of the reasoning complexity, and access to the full ontologies for download. In OntoSearch2, the focus is on providing fast and scalable query access to the ontologies, while OntoSelect intends to support ontology-based annotation, in particular by providing language related information about the collected ontologies, as well as text-based search for ontologies.

Domain coverage: Is the ontology library designed to serve the needs of users in a particular domain? Most of the libraries in our survey are not limited to a specific domain (Table 1), several, however, are domain-specific. For instance, BioPortal focuses on biomedical ontologies, while oeGov contains the ontologies in the e-Government domain. The domain addressed by a library determines not only the content, but also the set of features as it is often custom-tailored to a particular user community.

As we can see from Table 1, the domain of biomedicine clearly dominates among the domain-specific libraries. This finding is not surprising, since biomedical informatics has adopted ontologies more readily and has developed and used more ontologies than most other fields [55]. Indeed, the field even has a specific conference focussing on the ontologies in the domain—the International Conference on Biomedical Ontology, ICBO [4]. However, with ontologies becoming more and more popular in other domains such as e-Government, this

Library	**Domain**	**Intended Use**
BioPortal	Biomedical	General ontology access and community evaluation
Cupboard	General	General ontology access and community evaluation
OBO Foundry	Biomedical	Collaboration space to develop a coherent set of ontologies
oeGov	e-Government	Access to a list and description of ontologies
OLS	Biomedical	Programmatic access and search capabilities
ODP	General	Support for ontology design through pattern reuse
OntoSelect	General	Use of ontologies for text annotation
OntoSearch2	General	Search and query of ontology content
ONKI	Business, geography cultural heritage,	General ontology access
Schema-Cache	General	Access to a list and description of ontologies
TONES	General	Tool testing and profiling

Table 1
Domain coverage and intended use of an ontology library. The second column describes the domain of the ontologies in the library. "General" domain indicates that there are no domain restrictions on the content. The third column describes the intended use of the library

dominance of biomedical ontologies and their libraries might be only temporary.

6 Library Content

The core aspect of any ontology library is its content. Several characteristics determine the content of the library, including how the content gets into the library, who controls the addition of the content, which quality controls are applied before an ontology is added to the library, and what type of content does the library include.

6.1 Collecting Ontologies

The methods for collecting and updating ontologies range from *manually adding ontologies* to the collection, to *federating* content from other libraries, or from the Web.

Table 2 shows a summary of the methods that ontology libraries use to collect their content. Most libraries allow manual inclusion of ontologies. In some libraries, such as oeGov and TONES, the prerogative of deciding what to include lies with the library administrators. There is no direct mechanism for the broader community of users to contribute ontologies directly to these libraries. In most libraries in our survey, however, users (who usually need to register with the library) can submit ontologies to be added to the collection through a dedicated user interface. In the case of Cupboard, each user can manage her own ontology space (or set of ontology spaces) that represents the set of ontologies she is controlling, with each ontology space contributing to the overall library.

OntoSelect is an example of a library that collects ontologies automatically from the Web and OLS constructs its collection by pulling it from the OBO Foundry. As the result the OLS content is a subset of the OBO Foundry content.

Several libraries use a hybrid approach for collecting ontologies, both manual and automatic. For example, in addition to user-submitted ontologies, BioPortal federates the ontologies from the OBO Foundry collection, and Cupboard federates ontologies from ODP, by pulling them regularly from their original repositories.

6.2 Level Of Gatekeeping

Closely related to the question of how the ontologies get into the library is the level of gatekeeping that the library maintains. The level of gatekeeping is determined by the criteria that are applied to decide whether an ontology should be in the library, the process for this initial assessment, and the workflow for making the decision (Table 2).

On the one end of the spectrum are libraries, such as Cupboard, and SchemaCache that do not impose any quality control or any limitation on the ontologies that are included in the library. Both, however, require that the ontology is syntactically correct and can be processed by an RDF or OWL parser. In BioPortal, any registered user can submit an ontology and it immediately appears in the collection. There is a minimal level of curation, with regular review of the ontologies by BioPortal administrators, who occasionally need to remove ontologies that are clearly "toy" ontologies or are clearly out of scope. ODP has a two-tiered approach, originating from its purpose of providing good, reusable practices in ontology design. In ODP, registered users submit ontology design patterns (ODPs) and add then to the list of ODPs to be validated by the "quality committee." The quality committee then evaluates and validates the submitted ODPs and adds them to the official "catalogue." However, both the list of submitted and validated ODPs are visible to all the users. At the other end of the spectrum is the OBO Foundry library (and, consequently, OLS, which federates from it), which has an explicit editorial policy that lists specific quality requirements for ontologies to be included in the OBO Foundry collection. The requirements range from syntactic ones, such as documentation and unique ID across the whole collection, to the level of usage, requiring a plurality of users, to content: if authors of an ontology want the ontology to be included as an OBO Foundry candidate, they must work to remove any overlap with other ontologies that are already there. The OBO Foundry does not formally define the workflow for determining confor-

Library	**Collection**	**Gatekeeping**
BioPortal	Submitted by registered users; federation from OBO Foundry; import from CVS and SVN	Manual curation and automatic syntax validation
Cupboard	Submitted by registered users; import form other libraries	Automatic syntax validation
OBO Foundry	Submitted by users	Developers must conform to OBO Foundry principles; an editorial committee must
		recommend inclusion
oeGov	Added by administrators through a blog post	Performed by administrators
OLS	Federated from OBO Foundry and added by administrators	same as OBO Foundry
ODP	Submitted by registered users	Validation by committee and registered users
OntoSelect	Crawling; submitted by users	Automatic syntax validation
OntoSearch2	Submitted by administrators and users	Automatic syntax validation
ONKI	Administrators	Manual curation and automatic syntax
		validation
Schema-Cache	Submitted by users (no need to register)	Automatic syntax validation
TONES	Submitted by administrators	Performed by administrators

Table 2
Content collection and gatekeeping mechanisms. The second column lists where the ontologies in the library come from and who can submit them. The third column describes the validation that is performed when the ontology is submitted to the library and the gatekeeping rules that determine what ontologies the library can include.

mance to these principles. However, there is a group of dedicated curators who engage the authors in mailing list discussions and maintain the listing on the library site. These curators informally decide on the inclusion of the ontologies. Finally, the libraries where administrators are the only ones making decisions on what to include, usually do not have well defined gatekeeping policies and the administrators decide on an ad-hoc basis which ontologies to include, usually based on the purpose and scope of the library (Section 5).

6.3 Content Characteristics

A number of characteristics of an ontology library are easy to evaluate and they can often give a good idea of the amount and quality of content in the library, as well as of the type of content that is allowed.

6.3.1 Size

Size is a seemingly straightforward measure that we can use to assess coverage of an ontology library. Note, however, that the notion of what an ontology is differs from one library to the next, and the way libraries measure the number of ontologies that they contain is not consistent throughout the set of libraries in our survey. For example, in providing the number of ontologies that it includes, the OBO Foundry counts only the latest version of each ontology; Cupboard counts each version as one ontology (see Appendix for the discussion of version management). We used the following approach to determine the numbers of ontologies in each library that we summarize in Table 3: if the ontology library reports the number of ontologies that it includes, we used that number. Otherwise, we determined the number of ontologies ourselves by studying the listing of the ontologies that the libraries provide. We also indicate if, based on our observations over a period of a few months, the number of ontologies is currently growing, or if it is staying relatively stable.

As can be expected, general ontology libraries that do not restrict their content to a specific domain tend to contain more ontologies than domain specific ones, with the exception of relatively "young" and still growing systems such as Cupboard.

It is also worth noting that OntoSelect clearly stands out in terms of the number of ontologies it contains. This difference stems from the way OntoSelect gets its content: many of the ontologies are added as the result of a crawl that looks for specific types of documents. Thus, on the one hand, there is no gatekeeping, and on the other, there is no implicit temporal filter as in other ontology libraries: if someone needs to add an ontology to the library manually (even if the library takes all comers), this ontology is unlikely to be very outdated, whereas an ontology picked up in a crawl may have been unused for a while.

6.4 Types Of Content

Ontologies themselves are, naturally, the focal point of the content that the ontology libraries provide. However, a number of libraries in our survey maintain additional content that links ontologies together. We focus on two types of the additional content: (1) mappings between ontologies; and (2) ontology-level relationships.

Ontology Mappings

In most ontology libraries, ontologies inevitably overlap in coverage. Some libraries include mappings between ontologies. In this context, a mapping is any explicitly asserted relationship between entities in different ontologies.

Of the libraries in our survey, only BioPortal, Cupboard, and the OBO Foundry inlude ontology mappings. BioPortal allows users both to create mappings interactively and to upload large sets of mappings created elsewhere [47]. Users can search mappings and download a set of mappings, filtered according to their specification (e.g., all mappings between two specific ontologies created by a specific user). BioPortal also stores extensive metadata on mappings, including details on how the mapping was created, as well as comments that other users can add to the mapping.

Cupboard relies on the Alignment Server [38] to provide ways for users to create and store mappings [32]. The Alignment Server implements a set of services for the storage, manipulation and export of ontology alignments (i.e., sets of mappings). It also includes a library of ontology matching methods which users can invoke to create new mappings between ontologies automatically. It stores simple metadata about the alignments and can export the alignments to a variety of formats (OWL, C-OWL, SWRL, etc.).

There are also other repositories that are dedicated specifically to storing mappings between ontologies, such as the OntoMediate system [28].

Ontology-Level Relations

Ontologies are not isolated artifacts: they may be related to one another, with the relations ranging from simple import to modularization, incompatibility,

Library	**Number of ontologies**	**Dynamics**
BioPortal	270	Growing
Cupboard	150	Growing
OBO Foundry	86	Stable
oeGov	31	Growing
OLS	79	Stable
ODP	125	Growing
OntoSelect	1530	Stable
OntoSearch2	200	Stable
ONKI	78	Stable
Schema-Cache	157	Stable
TONES	231	Stable

Table 3
Size of the libraries. The second column lists the number of ontologies in each library. The third column indicates whether this number is stable or growing.

and so on [22]. The mappings that we described in the previous section are entity-level mappings. Here we focus on ontology-level relations. The most common ontology-level relation between ontologies is dependency, supported through the `import` relation in OWL. Surprisingly, only a few of the ontology libraries we have studied provide such information and the ability to navigate through the ontologies using this relation (Cupboard).

Versioning is another such relation commonly appearing in ontology libraries. In the Appendix, we describe more precisely the mechanisms used in some of the ontology libraries to keep track of the different versions of each individual ontology, allowing to retrieve its whole history, or just the most up-to-date version.

Some libraries also provide the ability to share views or subsets of ontologies. For instance, the library can have a large ontology, such as the Foundational Model of Anatomy (FMA), as one of its entries. FMA is widely used in biomedical ontology community, but it is extremely large (more than 80K classes). Thus, many users have created custom-tailored subsets of FMA. For instance, a subset may be a specific subtree of an ontology [57]. BioPortal, for example, includes such views or subsets as part of its collection, allowing users to reuse the customizations of ontologies that others have created.

Finally, ODP also provides a set of simple, predefined relations that can be specified within the metadata of an ontology (or an ontology design pattern), including ontology links such as *Extracted From* or *Specialization Of*. The use of such ontology relations is still, however, very limited.

While researchers have identified other types of ontology-level relations [22], today's libraries do not support navigation of their collections through the relationships other than the ones we have described in this section.

6.5 Ontology Metadata

Ontology metadata is an essential part of many ontology libraries. The type of metadata that one can often find in an ontology library includes ontology name and domain; who created it and when; versioning policy; licensing policy; additional links and references. Elements of metadata are often directly attached to the ontology, for example, through ontology properties in OWL. Ontology libraries can include mechanisms to extract and integrate these metadata, and use it to represent and store additional elements directly within the ontology. However, the metadata that ontology authors usually include as part of the OWL file is often rather minimal and contains only the most essential information, such as the ontology authors, its name and description.

The libraries where ontology authors can submit their ontologies usually include a form-based interface that allows authors to provide the pertinent metadata about the ontology and its provenance information.

Because there is no standard format to represent metadata about ontologies, there is no uniformity in metadata support among libraries. There are emerging schemes for representing such metadata, with the Ontology Metadata Vocabulary (OMV) as one of the more extensive schemes for that [43]. However, OMV has not been tested yet and others have reported on the need to extend it when applied in a large-scale ontology library [52]. BioPortal and Cupboard currently support this format.

Other libraries do not provide any structured metadata (OLS, OntoSearch2), while some implement ad-hoc representations, either machine processable, API-based or only human-readable. For example, ONKI only provides a textual description of the ontologies it contains, but makes use of additional metadata (subject, status, structure, publisher) for browsing the repository, and oeGov gives basic information about the ontologies (description, namespaces) in a manually built table. The OBO Foundry, which uses sourceforge, stores metadata about ontologies in a separate XML file that is available in a specific place in the sourceforge repository and ODP uses the form mechanism provided by (Semantic)MediaWiki. OntoSelect and TONES show basic features extracted from the ontology files (such as number of entities, dependencies, languages, labels and expressivity of the underlying description logic). Schema-Cache provides an RDF description of the ontologies it indexes, based on a variety of generic vocabularies (DublinCore, FOAF, etc.).

7 Key functions

In this section, we compare the key functions that the ontology libraries provide to their users. Specifically, we focus on the way the libraries enable users to find and evaluate the ontologies, to browse them, and to access them programmatically.

7.1 Search

Search mechanisms are the core features of ontology libraries. They define the way ontologies are found, navigated and discovered in potentially large collections. While, at first glance, the support for such features can appear quite straightforward, we observed a lot of variation in the way different ontology libraries implement them, along the following dimensions:

Search within ontologies enables users to find elements in a specific ontology (classes, properties, individuals).

Search across ontologies provides a global search for terms within all ontologies in a library. Some libraries allow users to select a specific set of ontologies to search.

Structured search enables users to query for ontologies or elements of ontologies that have specific properties. For example, such search can be implemented through the use of a SPARQL query engine.

Advanced search correspond to the similar set of options that can be provided to many search engines to refine the search space.

Table 4 provides a summary of the search capabilities for the libraries in the survey. Some libraries provide search only for ontology names in the list—the

OBO Foundry and TONES are in this category. ODP and oeGov go a step further, enabling search through the metadata of the ontologies, using the underlying wiki and blog search mechanisms, respectively, to search through ontology descriptions. Schema-Cache and ONKI provide the simple form of search for ontologies, while OLS enables users to search preferred names and synonyms of ontology terms. BioPortal and Cupboard provide advanced search capabilities within and across ontologies, enabling users to specify where in the ontologies to search for the keywords, and which ontologies to include in the search. OntoSearch2 provides SPARQL queries enabling users to pose detailed structured queries to the libraries in the collection. When provided, advanced search mechanisms generally consist of options to restrict the search space (e.g., search within a category in BioPortal, only for certain types of entities in Cupboard, only certain languages in OntoSelect). If available, the mechanism for searching inside ontologies also benefits from these search options (BioPortal, Cupboard). Finally, OntoSelect enables users to find ontologies that are relevant to the main topic of a particular webpage or wikipedia article.

7.2 Browsing

While a number of ontology libraries focus on providing a listing of ontologies (TONES, oeGov, ODP) and, perhaps, a uniform way to download them (e.g.,

Library	**Search metadata**	**Search within ontology**	**Search across ontologies**
BioPortal	Yes	Yes, with autocomplete	Advanced search (attributes, restrict to category); autocomplete
Cupboard	Yes	Advanced search	Advanced search
OBO Foundry	Yes	No	No
oeGov	Blog-based	No	No
OLS	No	Yes, terms and terms IDs; exclude obsolete terms	Yes
ODP	Wiki-based search on terms ids	No	No
OntoSelect	No	Topic-based search	Advanced search (restrict to format, language)
OntoSearch2	No	Yes	Keyword-based; structured search (query-based)
ONKI	No	Yes, with autocomplete	keyword-based
Schema-Cache	No	Keyword-based	keyword-based
TONES	Search only for ontology name	No	No

Table 4
Search Mechanisms supported by ontology libraries. The second column indicates whether the library provides search capabilities for ontology metadata. The third column describes the search capabilities within a single ontology that the user specifies. The final column lists the search capabilities across ontologies (such as searching for a term in a set of ontologies or all the ontologies in the library).

TONES), others provide the additional capabilities for browsing the content of the ontology collection (Table 5).

Ontology browsers can take many different forms. In general, an ontology library should provide a way to browse a set of ontologies, whether it is the entire collection or selected sub-sets. Here, important distinguishing aspects between the browsing features provided by ontology libraries include:

Navigation criteria correspond to the elements that are used to order and link the ontologies with each other for the purpose of supporting navigation in the collection. A few systems allow to navigate through the ontologies by following links they share (Cupboard, ODP and Schema-Cache). In particular, Cupboard and BioPortal provide an interface that allows users to navigate from ontology to ontology, following shared entities, import links, or mappings between them. ONKI provides faceted browsing of its list of ontologies, including browsing by subject, publisher, and type of ontology.

Sub-spaces can also be provided, allowing to browse not only the entire collection, but also pre-defined subsets (e.g., Cupboard).

Browsing ontologies is a feature sometimes provided, allowing users to browse the content of ontologies within the library system, and not only the ontology collection. Browsing inside an ontology can take different forms: BioPortal, for example, provides a variety of ways to visualize ontologies. ONKI and Cupboard display a hyperlinked representation of the entities, which users can navigate. OLS provides a tree representation of the ontology, which gives access to the complete description of each of the entities it contains.

7.3 Selecting and evaluating ontologies

One of the primary functions of many ontology libraries is to assist ontology users in selecting the ontologies that they need for their applications. The

Library	**Navigation Criteria**	**Browsing ontologies**
BioPortal	Ordered by name, author, version, etc.	Yes
Cupboard	Ordered by date , linked through references	Yes
OBO Foundry	Ordered by name, domain, etc.	No
oeGov	Ordered by publication date	No
OLS	Ordered by name	Yes
ODP	Ordered by name, domains, etc.	No
OntoSelect	Ordered by metrics	No
OntoSearch2	No navigation available	No
ONKI	Faceted browsing (subject, structure, publisher, etc.)	Yes
Schema-Cache	No specific ordering	Yes
TONES	Ordered by metrics	No

Table 5
Browsing mechanisms supported by ontology libraries. The second column describes the different ways in which the users can order and navigate the *listing* of the ontologies in the library. The second column contains “yes” if the library provides browsing capabilities for the content of the ontologies themselves and “no” of it does not.

libraries use different approaches in helping users assess the size, complexity, and scope of the ontologies and evaluate their applicability in various circumstances (Table 6).

Computable ontology metrics: There are a number of metrics that tools can compute on an ontology when it is uploaded to the library. The metrics can include simple quantitative information such as the number of classes, properties, restrictions (e.g., BioPortal, Cupboard, and OntoSelect all provide such metrics). Other metrics include the import relationships between ontologies, the DL sublanguage that an OWL DL ontology falls into, or the auditing information, such as the number of classes with no documentation, authors who have contributed to the ontology, and so on. For example, the TONES repository provides more than 30 metrics on each ontology, including the number of times the ontology uses different OWL features, such as disjointeness, the number of elements of different types, the number of symmetric, reflexive, and transitive properties, and other OWL features. BioPortal adds auditing metrics that indicate how well an ontology conforms to best practices of ontology design [48], such as the number of classes with a single subclass, classes with large number of direct subclasses (often indications of poor modeling choices), and number of classes with no textual definitions. OntoSelect focuses on metrics that related to the use of natural languages in ontologies.

Comments and reviews: As elsewhere on the web, some ontology libraries are harvesting the power of the community to augment their functionality. For instance, in addition to submitting ontologies to the library, users may be able to comment on ontologies or their components, describe their ex-

Library	Metrics	Comments and reviews	Ranking
BioPortal	Syntactic and conformance to best practices	Ontology-level and class-level	
Cupboard	Syntactic, DL Expressivity	Ontology-level	TF.IDF
OBO Foundry	Syntactic	Ontology-level (CVS term tracker for each ontology)	
oeGov		Ontology-level (blog-based)	
OLS			
ODP		Ontology-level	
OntoSelect	Syntactic		Metric-based
OntoSearch2			
ONKI		Ontology- and entity-level	
Schema-Cache	Syntactic		
TONES	Syntactic; DL Expressivity		

Table 6
Ontology evaluation mechanisms. The second column lists the types of metrics that the library provides. Most libraries provide syntactic metrics such as the number of classes, properties, and axioms. Empty cells in this table indicate that there is no evaluation mechanism present. The third column lists the types of user feedback and evaluation that the library collects. It highlights whether users can provide the feedback at the level of ontology as a whole ("ontology-level") or at the level of a single class ("class-level"). The final column links the mechanism, if any, that the library uses to rank the ontologies.

periences with using ontologies, make suggestions and requests for changes. The ontology libraries in our survey differ in the types of community-based comments and reviews that they have.

- *Ontology level comments, ratings and reviews:* Users can provide ratings and reviews on ontologies as a whole, usually indicating how good they think an ontology is, how appropriate it is for a specific task, what is the level of support provided by developer, which parts of the ontology are developed better than others, and so on. The ontology-level reviews are usually geared towards helping other users to evaluate the ontology and its appropriateness for a task. The libraries may differ on who can provide reviews, whether or not they support the discussion of reviews and ratings of the reviews themselves (for example, enabling users to answer the question *"Was this review helpful to you?"*). Developers of some ontology libraries, such as Cupboard and BioPortal, have proposed integrating web of trust mechanisms and other ways of aggregating reviews. The OBO Foundry takes a different approach in helping users assess which reviews to trust: it has the "editorial board" model, where a group of experts evaluated the ontologies.
- *Class-level comments and notes*: The libraries that provide detailed visualization of ontologies (e.g., BioPortal, ONKI), may allow users to comment on specific classes in ontologies. One can imagine a variety of uses for these types of comments and indeed a structure to them, such as passing the feedback on the classes to the ontology authors, suggesting changes and corrections, requesting new items, discussing class definitions among a group of contributors, and providing additional information about a class, such as references, images, or supporting documentation. The OBO Foundry, for example, includes an issue tracker that allows users to request new terms. BioPortal supports notes on classes. The notes usually have provenance information, such as who created them and when.

It is worth mentioning here however that, while several libraries provide mechanisms for users to comment on and review ontologies, these mechanisms are generally scarcely used. In BioPortal and Cupboard for example, there are no reviews for a large majority of ontologies, and it is rare for an ontology to have more than one review (e.g., there is only one ontology with two reviews in BioPortal at the time of writing). In BioPortal, the main use of the comments feature for users is to request new terms. Indeed, BioPortal provides a form of structured notes to create such term requests [21].

Ontology ranking: In addition to providing "static" information about ontologies, such as metrics and reviews, ontology libraries may also provide ranking of ontologies, usually in response to a user query. The ranking indicates which ontology is the most appropriate for the query, for example, by covering better the keywords specified in the query, using standard information retrieval measures such as TF.IDF (term frequency-inverse document frequency [56]) in Cupboard.

As with other metadata, standard representation for metrics, reviews, notes, and ranking can enable sharing of the metrics that one tool computes with other libraries. For instance, metric information can be part of the ontology metadata represented in OMV. Then, sharing the metadata this way enables sharing the evaluation data as well.

7.4 Programmatic Access to Ontologies

An ontology library may be geared primarily for interactive browsing or it may provide programmatic access to its content. Programmatic access enables application developers to use the content of the ontology library "live" in their applications. Most ontology libraries that do provide programmatic access provide it through Web service protocols, such as SOAP and REST. The Web services usually provide access to different types of content, including metadata about the ontologies, search functionalities, ontology content, and query access, such as a SPARQL endpoint.

Table 7 summarizes availability of programmatic access to the content of ontology libraries. BioPortal, Cupboard and ONKI provide REST service access to the content. In all three cases, there are services to access most of the content that the user interface presents, including the metadata about ontologies in the library, the ontology content, details of specific concepts, and, in the case of BioPortal and Cupboard, mappings between ontologies. OLS provides access to its content through SOAP services. BioPortal and Cupboard also provide REST services to add content to the library. ONKI and OLS provide service only to look up the information but not to alter it or to add new information. A number of libraries provide SPARQL endpoints.

Library	**Web service access**	**SPARQL endpoint**	**Content available**	**Read or Write**
BioPortal	REST	No	Ontology content, metadata, mappings	Read and write
Cupboard	REST	Yes	Ontology content, metadata, mappings	Read and write
OBO Foundry	No	No		
oeGov	No	No		
OLS	SOAP	No	Ontology content	Read
ODP	No	No		
OntoSelect	No	No		
OntoSearch2	No	Yes	Ontology content	
ONKI	REST	No	Ontology content, metadata	Read
Schema-Cache	REST	Yes	Ontologies and metadata	Read and write
TONES	REST	No	Download	Read

Table 7
Programmatic access, Web services, and APIs The second column lists the types of Web-service access, if any, that the library supports. The third column indicates whether or not the library provides a SPARQL endpoint for its content. The fourth column lists the type of content that is accessible through the API. The column is blank if the library does not provide Web service access. The final column indicates whether there are Web services to update the library content ("write") or only to query it ("read"). The column is blank if neither read nor write is available.

8 Discussion

In this paper, we have presented a comprehensive survey of active ontology libraries, their similarities and distinctions. We now step back from the detailed analysis of each property to consider the practical implications from this survey for ontology users and Semantic Web practitioners, to discuss what the survey tells us about the current state of the ontology-library technology, and what should be the next steps in developing this technology.

8.1 Different Types of Ontology Libraries

One of the key conclusion that we can draw from our survey is that the set of applications that we describe using the generic term *ontology library* cover several different types of systems, with different goals and target audiences. This large variety of features and the numerous differences in approaches can be confusing to a user new to the area. Indeed, while on the surface all ontology libraries appear similar, there is little in common between, for example, the OBO Foundry and OntoSearch2. The one common feature is that all the libraries that we discussed provide means for discovering and accessing a set of ontologies—they all serve as *ontology repositories*. By collecting a number of ontologies in a single place, these systems facilitate the task of discovering ontologies and selecting ontologies that the user needs. They all provide some ways to access the information in the ontologies or to download them, and different mechanisms for finding relevant ontologies.

Besides the repository function, we can distinguish the following broad categories of ontology libraries, with each category targeting a different set of users: (1) curated *ontology directories*; (2) *ontology registries* that serve as ontology-publishing platforms for users; and (3) *application platforms*, that provide programmatic access to ontology content. While these categories have distinct descriptions, several systems are designed to handle several of these generic usage scenarios.

We now describe these categories in more detail.

Libraries of curated ontologies (ontology directories). Perhaps surprisingly, many of the ontology libraries we have studied are not general purpose, domain-independent collections of ontologies, but are explicitly focusing on a particular field and specific usage scenarios. This category includes such systems as the OBO Foundry, ODP, TONES, oeGov, and ONKI. A particular aspect of these libraries is that they put an emphasis on the content of the carefully curated collection of ontologies they provide, rather than on the features of the system. Indeed, most of them are populated manually by the administrators of the system, without the possibility for other users to contribute. In this category, where users can contribute their own content, the systems include strong validation and quality-assurance mechanisms in their process. The goal of these *ontology directories* is therefore to provide *reference ontologies* in a particular domain (biomedicine, e-government, ontology design) with a certain level of guarantee on the validity of these ontologies.

Ontology-publishing platforms (ontology registries). Another role an ontology library can play, in contrast with the previous category, is to provide a platform for *ontology developers* to publish and share their ontologies in a convenient, valid, and efficient way. As a larger variety of practitioners create more and more ontologies, these registries provide platforms for them to publish their ontologies and to make them available to others. The availability of these platforms is crucial in ensuring that ontologies can be distributed and reused across communities, without the ontology developers having to set up their own infrastructure to support them. This type of systems thus focus on providing mechanisms for these ontology developers to *register* and *publish* their ontologies easily, sometimes providing specific spaces for different users or different categories of ontologies. Such systems are generally open for any user to contribute (as long as the users are registered) and apply only limited validation mechanisms on the submitted ontologies. These systems put a particular emphasis on the features that support the management of ontologies, including the ability to keep track of versions of ontologies and to include mappings between them. However, while such features seem crucial, considering the growing popularity of the Semantic Web, from the libraries in our survey, only BioPortal and Cupboard seem to be designed to act as *ontology registries*.

Libraries providing technical infrastructures for re-using and using ontologies (application platforms). A final category includes ontology libraries that are dedicated to supporting application developers in using the content of the ontologies to build applications. These applications will often use multiple ontologies and access them directly through the library. We called such systems *application platforms* as they focus on providing a technical infrastructure to support programmatic access to information, both *about* the ontologies and the *content* of the ontologies. This access is provided through querying mechanisms, as well as search and exploration services. Providing such support is a key component of several systems, such as Cupboard, BioPortal, SchemaCache, and OntoSearch2. In this category more than in others, the scalability, robustness, and performance of the underlying technological infrastructure are crucial to guarantee reasonable access for a possibly large number of demanding applications.

Table 8 provides an overview of these categories and shows where the systems in our survey fall. Understanding what are the different categories and what is the functionality of the systems in each category enables Semantic Web developers to understand which ontology library they need to use for their specific task. More specifically, it helps formulating the criteria to choose a library that would best fit a particular need. For example, an *ontology developer* who wants to publish an ontology could ask the following questions:

- *Is the goal of publishing the ontology to support its broad adoption and reuse?*

In such cases, the focus for the ontology library system should be on the features it provides to search and browse ontologies, as well as additional information, and evaluation to support users in finding and adopting the ontology. The ontology developer should consider in priority libraries that fall into the *ontology registries* category.

- *Is the goal of publishing the ontology to ensure some level of endorsement from a particular community?* While of course this question relates to the previous one (i.e., endorsement might be thought as a way to support adoption and reuse), here the emphasis should be on libraries that provide an appropriate level of validation of the ontologies they include, and that preferably target the considered community, therefore favouring *ontology directories.*
- *Is the goal of using a library to provide shared online access to the ontology?* Depending on whether the emphasis is on access by humans or applications, different features might be considered. In the case of applications, the programmatic access provided by *application platforms* seems an important criterion to consider. However, in the case collaboration between developers and users of the ontology is important, the features of application platforms are still relevant as they allow for the ontology to be easily integrated in various software environments. In such cases, libraries combining features from ontology registries and application platforms might be considered.

Similarly, an *ontology user* looking for an appropriate ontology library might consider the following questions:

- *Is it important for ontologies to be validated by a given community?* The level of validation and endorsement by a particular community for an ontology can be very important in scenarios where the user of the ontology library cannot assess the adequacy of the ontology herself, and when the usage of the ontology should comply with some kind of community practices. *Ontology directories* are naturally more appropriate in such cases, especially if the ontologies obtained from the library should all relate to the same domain.
- *Would ontologies relate to one particular domain, one particular format, or will heterogeneous ontologies be considered?* In cases where the ontologies to be obtained from the library might come from a variety of sources, covering different domains and formats, the generic search and browsing facilities provided by *ontology registries* have an advantage, especially in cases where additional content is available, such as ontology metadata and mappings.
- *Would ontologies be used in applications and integrated with other software environments?* It is often assumed that ontologies would be found in a library and downloaded to be used locally. However, in scenarios where the ontologies from the library should be integrated in applications and other software environments, the programmatic access provided by *application platforms* to both the content of the ontologies and to features such as search and metadata retrieval represent a natural advantage.

	Ontology Directory	Ontology Registry	Application Platform
BioPortal	∼	√	√
Cupboard		√	√
OBO Foundry	√		
oeGov	√		
OLS	√		
ODP	√		
OntoSelect		∼	
OntoSearch2			√
ONKI	√		√
Schema-Cache		∼	√
TONES	√		∼

Table 8
Classification of ontology libraries in different categories √ means "belongs to the category", ∼ means "provides some of the features related to this category". Blank means "does not belong to the category."

We hope that the details that our survey provides will help both ontology developers and users navigate the landscape of ontology libraries using the questions above as starting point.

8.2 Open Issues

We outlined the key features that today's ontology library provide, and gave an overview of the range of available systems. However, our analysis also highlights issues that are not sufficiently addressed by current systems, missing features and, broadly, gaps in this landscape that should inform the developers of ontology libraries and researchers interested in ontology publishing.

From a broad perspective, a question to be asked is whether ontology libraries are successful. We argued that ontology libraries should support the activity of ontology reuse, which involve facilitating the discovery and integration of ontologies. There is no data yet to show that publishing ontologies in the ontology libraries is actually an effective way to support users in finding and reusing ontologies. Indeed, the current proliferation of systems and approaches might actually be a symptom of the lack of efficient methods to address the problem of discovering relevant ontologies. Some of the systems rely on the metadata attached to ontologies to help users in understanding their content, and assessing their relevance. Others provide visualizations of the whole content of an ontology, or of a meaningful summary. However, a number of questions cannot be directly answered using these tools: "where has an ontology been used before?" or "is this ontology compatible with mine?"

Looking at the current state of ontology libraries, one might therefore ask whether or not they are actually relevant. Indeed, it can be easily argued that most ontology practitioners reuse ontologies without necessarily using ontology libraries. However, this fact alone does not necessarily imply that ontology libraries are irrelevant, but rather that both the systems and the practices for ontology reuse are not mature enough yet. Ontology reuse is becoming a more difficult task, with the proliferation of ontologies that constantly appear on the Semantic Web. Ontology practitioners often rely on their social network to find and select ontologies to reuse. However, in addition to not being very scalable, this approach does not address new ontology practitioners who might not share any connection with ontology developers in their community. Ontology libraries have a role to play, but as the social practices of ontology reuse evolve, the systems also need to evolve to include these social practices. Indeed, it appears as an anomaly that most of the systems we have considered in our survey do not provide ways for ontology practitioners to connect to the developers and other users of ontologies, and none of them integrate the otherwise omnipresent social networking features that would enable more contemporary ways to engage in the communities related to ontologies.

To a large extent, there is a parallel between ontology libraries and online code sharing environment, especially in the open source world. In both cases, the goal is to publish online an engineering artifact so that others can reuse it. The two types of repositories address different types of users and use cases. For example, most users of source-code repositories use these repositories to *extend* or *reuse* the code. Many users who look for ontologies in the ontology libraries do not plan to extend them by writing their own ontologies (akin to their own code) but rather to use the exiting ontologies in their applications. [1] At the same time, looking at the features of mature source-code repositories highlights open issues that ontology-library developers must still address:

Documentation and Licensing Information. We can achieve a higher level of ontology reuse only if users can access the details on the conditions under which an ontology can be reused. We mentioned that several systems include metadata describing certain characteristics of ontologies such as their provenance. However, such metadata does not generally include characteristics that directly affect the possible reuse of the ontology. Licensing for ontologies in particular is an area that have been given little attention in all the systems we reviewed.

Access Privileges. In all the systems that we studied in this survey, all ontologies are available to all users. Because not all ontologies are available to everyone without restrictions—and this situation will be exacerbated as more commercial entities start developing proprietary ontologies—functionalities related to access control will become critical.

Interactions between Contributors and Contributions. As more libraries enable their users to review content, and to provide notes, mappings, and other metadata, they must address the interaction between the users and the content that they contribute. For instance, none of the libraries

[1] While a number of ontology projects use sourceforge to track requests, there are serious limitations to this approach [61] and a number of these communities are moving to BioPortal.

has clear policies on who can delete or change a note or a mapping that a user has created, who can change the metadata about an ontology and who can create mappings for ontologies. As libraries support different types of contributions, a larger number of these questions will have to be addressed.

Evolution of Ontologies. Ontologies on the Semantic Web evolve and authors publish new versions. Ontology libraries must maintain all the metadata and user-contributed content through this evolution process. Users add the metadata for specific ontology versions, and, in theory, any metadata can become invalid when a new version is published. For example, if a class definition changes, a mapping may become invalid; or a note, requesting a change to a class, could become irrelevant. Similarly, a review that indicates some problems with an ontology may no longer be adequate after the ontology has been fixed. At the same time, we do not want to invalidate all user-contributed content linked to an ontology once a new version of that ontology is uploaded: Our earlier research shows that only a small fraction, usually 1-4%, of ontologies changes from one version to the next [49]. Libraries that we surveyed take different approaches to this problem. For instance, BioPortal uses a hybrid approach: all metadata, such as comments, mappings, reviews, are attached to a specific ontology version. The same is true for reviews in ODP. However, the metadata also references the global ontology ID and the user interface exposes the metadata when users access a newer ontology version. ODP users can also link new versions of ontologies to the reviews that the version addresses.

To Edit or not to Edit? None of the ontology libraries in our survey provides a way for users to edit the ontologies themselves. However, many papers that describe these libraries talk about support for ontology editing as the natural next step in their evolution. Developers have not yet determined what are the workflows for such editing: Who should be allowed to edit? How frequently are the new versions created? What are the processes for reaching consensus and resolving conflicts? Indeed, is support for ontology editing even a reasonable feature to have in a library or should a library link to other tools that support ontology editing?

Finally, one question that none of the libraries is addressing yet is the following: how can ontology libraries interoperate? We have discussed eleven different systems, and we can clearly see many redundancies in these systems, both in their content and in their features. Besides the duplication of effort, it seems to be contradicting the very goal of ontology libraries that a user would have to use of several different systems to find ontologies to reuse, generally having to adapt to the specifics of each system and to the way to interact with them. Furthermore, we must develop ways to aggregate user-contributed content, such as comments, reviews, and mappings. The pool of contributors of such content is rather small, and it is unlikely that users would enter the same review, or the same piece of metadata, for the same ontology across all the available systems. Our community must develop recommendations specifying standard ways for users and tools to access, communicate and contribute to ontology libraries, creating a more homogeneous experience for users, and the possibility for ontology library systems to network and exchange content with one another.

9 Conclusions

There are different reasons to create an ontology library and many different reasons to use one. As a result, the number of such systems has grown dramatically, in a way that could appear confusing to ontology practitioners wishing to publish their ontologies, to find existing ones to reuse, or to exploit existing ontologies efficiently.

In this paper, we have presented a comprehensive survey on eleven currently available ontology-library systems. Our survey has demonstrated a large variability among the ontology libraries. The libraries' initial purpose and domain as well as projected usage significantly affect the type of content that the library contains and the set of key functional features that it provides. While it is still hard for users to understand which ontologies have been used more actively in applications and what are the trade-offs of using or extending specific ontologies, the ontology libraries that we surveyed are beginning to address these needs, giving users a place to start.

Acknowledgment

This research was supported in part by the National Center for Biomedical Ontology, under roadmap-initiative grant U54 HG004028 from the National Institutes of Health.

The authors of this paper are actively involved in the development of two systems in the survey: Cupboard and BioPortal.

References

[1] Gene Ontology - OBO Edit (including an OBO API). http://sourceforge.net/projects/geneontology/.

[2] The OWL API. http://owlapi.sourceforge.net/.

[3] The Protégé ontology editor and knowledge acquisition system. http://protege.stanford.edu/.

[4] International Conference on Biomedical Ontology (Nature Proceedings). http://precedings.nature.com/collections/icbo-2009, 2009.

[5] MMI ontology registry and repository. http://mmisw.org/orr/, 2009.

[6] OntologWiki: Open Ontology Repository. http://ontolog.cim3.net/cgi-bin/wiki.pl?OpenOntologyRepository, 2009.

[7] Mediawiki.org. http://www.mediawiki.org, 2010.

[8] The NeON Project. http://www.neon-project.org, 2010.

[9] Ontologies for e-government. http://oegov.org/, 2010.

[10] Ontology Design Patterns.org (ODP). http://ontologydesignpatterns.org, 2010.

[11] OwlSight. http://pellet.owldl.com/ontology-browser/, 2010.

[12] Protégé ontology library. http://protegewiki.stanford.edu/wiki/Protege_Ontology_Library, 2010.

[13] Schema-cache. http://schemacache.test.talis.com/, 2010.

[14] Schemaweb.info. http://www.schemaweb.info/, 2010.

[15] Semantic Web for Earth and Environmental Terminology (SWEET). http://sweet.jpl.nasa.gov/ontology/, 2010.

[16] Sourceforge.net: Find and develop open source software. http://sourceforge.net/, 2010.

[17] Talis platform. http://www.talis.com/platform/, 2010.

[18] TONES Ontology Repository. http://owl.cs.manchester.ac.uk/repository/, 2010.

[19] Wordpress: Blog tool and publishing platform. http://wordpress.org/, 2010.

[20] M. N. Ahmad and R. M. Colomb. Managing ontologies: a comparative study of ontology servers. In *ADC '07: Proceedings of the Eighteenth Conference on Australasian Database*, pages 13–22, Darlinghurst, Australia, Australia, 2007. Australian Computer Society, Inc.

[21] P. R. Alexander, C. I. Nyulas, T. Tudorache, T. Whetzel, N. F. Noy, and M. A. Musen. Semantic infrastructure to enable collaboration in ontology development. In *International Workshop on Semantic Technologies for Information-Integrated Collaboration (STIIC 2011)*, Philadelphia, PA, USA, 2011.

[22] C. Allocca, M. d'Aquin, and E. Motta. Door - towards a formalization of ontology relations. In Jan L. G. Dietz, editor, *Proc. of the First Conference on Knowledge Engineering and Ontology Development*, pages 13–20. INSTICC Press, 2009.

[23] K. Baclawski and T. Schneider. The Open Ontology Repository Initiative: Requirements and Research Challenges. In *Workshop on Collaborative Construction, Management and Linking of Structured Knowledge at ISWC 2009*, Chantilly, VA, 2009.

[24] D. Berrueta and J. Phipps. Best practice recipes for publishing RDF vocabularies. Technical report, W3C Working Group Note, 2008.

[25] C. Bizer, T. Heath, and T. Berners-Lee. Linked data - the story so far. *International Journal on Semantic Web Information Systems*, 5(3):1–22, 2009.

[26] D. Brickley and L. Miller. FOAF Vocabulary Specification 0.9. Namespace document, FOAF Project, 2007.

[27] P. Buitelaar, T. Eigner, , and T. Declerck. OntoSelect: A dynamic ontology library with support for ontology selection. In *Proceedings of the Demo Session at the ISWC 04, Hiroshima, Japan, Nov. 2004*, 2004.

[28] G. Correndo, H. Alani, and P.R. Smart. A community based approach for managing ontology alignments. In *3rd International Workshop on Ontology Matching at ISWC 2008*, Karlsruhe, Germany, 2008.

[29] R. G. Côté, P. Jones, L. Martens, R. Apweiler, and H. Hermjakob. The Ontology Lookup Service: more data and better tools for controlled vocabulary queries. *Nucleic Acids Research*, 36(Web-Server-Issue):372–376, 2008.

[30] DAML. DAML Ontology Library. `http://www.daml.org/ontologies/`, 2000-2004.

[31] M. d'Aquin, C. Baldassarre, L. Gridinoc, S. Angeletou, M. Sabou, and E. Motta. Characterizing Knowledge on the Semantic Web with Watson. In *Workshop on Evaluation of Ontologies and Ontology-based tools, (EON)*, 2007.

[32] M. d'Aquin, J. Euzenat, C. Le Duc, and H. Lewen. Sharing and reusing aligned ontologies with Cupboard. In *Demo, International Conference on Knowledge Capture - K-CAP 2009*, 2009.

[33] M. d'Aquin and H. Lewen. Cupboard—a place to expose your ontologies to applications and the community. In *Proc. of European Semantic Web Conference, ESWC 2009*, volume 5554 of *Lecture Notes in Computer Science*, pages 913–918. Springer, 2009.

[34] M. d'Aquin and E. Motta. Watson, more than a semantic web search engine. *Semantic Web Journal*, 2011.

[35] M. d'Aquin, M. Sabou, E. Motta, S. Angeletou, L. Gridinoc, V. Lopez, and F. Zablith. Reusing knowledge from the Semantic Web with the Watson Plugin. In *Demo, International Semantic Web Conference, ISWC 2008*, 2008.

[36] L. Ding and T. Finin. Characterizing the Semantic Web on the Web. In *International Semantic Web Conference*, volume 4273 of *Lecture Notes in Computer Science*, pages 242–257. Springer, 2006.

[37] Y. Ding and D. Fensel. Ontology library systems. the key to successful ontology re-use. In *First Semantic Web Working Symposium*, pages 93–112, Stanford University, 2001.

[38] J. Euzenat. An API for ontology alignment. In *Third International Semantic Web Conference (ISWC 2004)*, pages 698–712, Hiroshima, Japan, 2004. Springer.

[39] A. Farquhar, R. Fikes, and J. Rice. The Ontolingua Server: a tool for collaborative ontology construction. *International Journal of Human-Computer Studies*, pages 707–727, 1997.

[40] Roy T. Fielding and Richard N. Taylor. Principled design of the modern web architecture. *ACM Transactions on Internet Technology (TOIT)*, 2(2), 2002.

[41] Gene Ontology Consortium. The Gene Ontology. `http://www.geneontology.org`, 2005.

[42] C. Golbreich, M. Horridge, I. Horrocks, B. Motik, and R. Shearer. OBO and OWL: Leveraging Semantic Web Technologies for the Life Sciences. *The Semantic Web*, pages 169–182, 2008.

[43] J. Hartmann, Y. Sure, P. Haase, R. Palma, and M. del C. Suárez-Figueroa. OMV – Ontology Metadata Vocabulary. In Chris Welty, editor, *Ontology Patterns for the Semantic Web Workshop*, Galway, Ireland, 2005.

[44] M. Hepp. GoodRelations: An ontology for describing products and services offers on the web. In *Proc. of the 16th International Conference on Knowledge Engineering and Knowledge Management (EKAW2008)*, pages 332–347. Springer, 2008.

[45] E. Hyvönen, K. Viljanen, J. Tuominen, and K. Seppälä. Building a national semantic web ontology and ontology service infrastructure — The FinnONTO approach. In *The Semantic Web: Research and Applications: Proceedings of the European Semantic Web Conference (ESWC 2008)*, pages 95–109, 2008.

[46] H. Lewen, K. Supekar, N. F. Noy, and M. A. Musen. TopicSpecific Trust and Open Rating Systems: An approach for ontology evaluation. In *Proceedings of the 4th International EON Workshop - Evaluation of Ontologies for the Web, located at the 15th International World Wide Web Conference WWW 2006*, 2006.

[47] N. F. Noy, N. Griffith, and M. A. Musen. Collecting community-based mappings in an ontology repository. In *7th International Semantic Web Conference (ISWC 2008)*, Karlsruhe, Germany, 2008.

[48] N. F. Noy and D. L. McGuinness. Ontology Development 101: A Guide to Creating Your First Ontology. Technical Report SMI-2001-0880 (also available as KSL Technical Report KSL-01-05), Stanford Medical Informatics, 2001.

[49] N. F. Noy and M. A. Musen. Ontology versioning in an ontology-management framework. *IEEE Intelligent Systems*, 19(4):6–13, 2004.

[50] N. F. Noy, N. H. Shah, P. L. Whetzel, B. Dai, M. Dorf, N. Griffith, C. Jonquet, D. L. Rubin, M. A. Storey, C. G. Chute, and M. A. Musen. Bioportal: ontologies and integrated data resources at the click of a mouse. *Nucleic Acids Research*, 10.1093/nar/gkp440, 2009.

[51] N.F. Noy and M. A. Musen. Promptdiff: A fixed-point algorithm for comparing ontology versions. In *Proc. 18th National Conf. Artificial Intelligence (AAAI-2002) (Edmonton, Alberta)*. AAAI Press, 2002.

[52] C. I. Nyulas, N. Noy, Michael Dorf, Nicholas Griffith, and M. A. Musen. Ontology-driven software: What we learned from using ontologies as infrastructure for software. In *5th International Workshop on Semantic Web Enabled Software Engineering at ISWC 2009*, Chantilly, VA, 2009.

[53] R. Palma, P. Haase, and As. Gomez-Pérez. Oyster: sharing and re-using ontologies in a peer-to-peer community. In *Proc. of the World Wide Web Conference, WWW 2006*, pages 1009–1010. ACM, 2006.

[54] J. Pathak, H. R. Solbrig, J. D. Buntrock, T. M. Johnson, and C. G. Chute. LexGrid: A framework for representing, storing, and querying biomedical terminologies from simple to sublime. *J Am Med Inform Assoc.*, 16(3):305–315, 2009.

[55] D. L. Rubin, N. H. Shah, and N. F. Noy. Biomedical ontologies: a functional perspective. *Briefings in Bioinformatics*, 9(1):75–90, 2008.

[56] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5), 1988.

[57] M. Shaw, L. T. Detwiler, J. F. Brinkley, and D. Suciu. Generating application ontologies from reference ontologies. In *AMIA Annual Symposium*, Washington, DC, 2008.

[58] B. Smith, M. Ashburner, C. Rosse, J. Bard, W. Ceusters W. Bug, L. J Goldberg, K. Eilbeck, C. J Mungall A. Ireland, N. Leontis, P. Rocca-Serra, A. Ruttenberg, S-A. Sansone, R. H Scheuermann, P. L Whetzel N. Shah, and S. Lewis. The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. *Nat Biotech*, 25(11):1251–1255, 2007.

[59] E. Thomas, J. Z. Pan, and Y. Ren. TrOWL: Tractable OWL 2 Reasoning Infrastructure. In *the Proc. of the Extended Semantic Web Conference (ESWC2010)*, 2010.

[60] E. Thomas, J. Z. Pan, and D. Sleeman. ONTOSEARCH2: Searching Ontologies Semantically. In *Proceedings of the OWLED 2007 Workshop on OWL: Experiences and Directions*, 2007. Online publication: http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-258/.

[61] Tania Tudorache, Natalya F. Noy, Samson Tu, and Mark A. Musen. Supporting collaborative ontology development in protege. In *7th International Semantic Web Conference (ISWC 2008)*, Karlsruhe, Germany, 2008.

[62] W3C. Ontaria ontology library. `http://www.w3.org/2004/ontaria/`, 2004.

A Additional Features

The appendix contains our discussion and comparison of the features of different libraries that might be of interest to more specialized audience. Specifically, we discuss version management for ontologies, technology infrastructure that underpins the libraries, the way the library integrates with the Semantic Web infrastructure in general, and other library features.

A.1 Version Management

Ontologies available on the Web (and in the libraries) often continue to evolve and their authors publish new versions of the ontologies. Thus, some libraries provide versioning mechanisms that enable users and developers to keep track of updates. Specifically, BioPortal, Cupboard, and the OBO Foundry store successive versions of the ontologies. The OBO Foundry uses the Sourceforge infrastructure for its version management. BioPortal and Cupboard implement their own version management. Cupboard always shows the latest version of an ontology in a given ontology space, while BioPortal provides two ways to access each ontology version: there is a version-specific access that resolves to a specific ontology version that the user request. There is also ontology-level access that resolves to the latest ontology version. For instance, an ontology-level URL to access a concept in an ontology will resolve to that concept's page in the latest version of the ontology.

BioPortal also provides a structural diff between successive versions of OWL ontologies in its collection, using the PromptDiff tool [51]. The OBO Foundry provides a textual diff through Sourceforge.

Many libraries do not consider version management important for their aims and store only a single version of each ontology. When a new version is added, it replaces the current version in the library. These libraries include oeGov, OLS, OntoSelect, OntoSearch2, TONES, and ONKI.

A.2 Infrastructure and Technology

Users do not often think of infrastructure and technologies that are at the core of a particular ontology library when deciding which library to use to search for or to publish ontologies. However, these elements can have an important influence on certain characteristics of the systems such as their scalability and robustness, in some cases making them inadequate for particular scenarios. The technology is particularly important for the libraries that must deal with a large number of ontologies or extremely large ontologies, such as some ontologies in the biomedical domain. For example, the Foundational Model of Anatomy has more than 80,000 concepts, the National Cancer Institute's Thesaurus is an OWL ontology with 50,000 classes, and numerous restrictions for each class, and the Gene Ontology has more than 20,000 classes. An ontology library that supports such large ontologies must use reliable and scalable infrastructure and robust technologies.

The following aspects can give us an insight into the technical infrastructure that ontology libraries deploy:

Underlying Infrastructure Components: Many ontology libraries use existing software components as part of their core infrastructure, such as software-engineering tools (e.g., CVS) or ontology tools (e.g., the Alignment Server, Watson, Protégé). The types of tools that are at the core of an ontology library often influence the types of features that the library can support.

Ontology Storage: Whether ontology libraries store the ontologies in their collections locally and how they do it, is critical to the performance of the system and the range of use cases that it can support. Not all ontology libraries actually store the ontologies in their collection. In some cases, they index the ontologies for searching or store only the ontology metadata.

Architectural Design: Most of the ontology library systems rely on a single server to support their operations. However, there can be different forms of distribution, for the purposes of performance, reliability or to separate different components of the system. While distribution adds to the complexity of the library system as a whole, it can also facilitate its ability to grow to a large number of ontologies.

Table A.1 summarizes the technological infrastructures underlying the ontology libraries in our study. Many of these systems reuse existing, well established components within their basic architecture. For example, the OBO Foundry is almost entirely based on Sourceforge [16], benefitting not only from the features provided by this system (version control, issue-tracker, etc.), but also from its maturity. Both oeGov and ODP reuse common content publishing systems (blogs in one case and wikis in the other). ODP implements many adaptations to and extensions of the original MediaWiki platform [7] to fit it to the task of ontology publication, while oeGov uses Wordpress [19] with almost no modification. Schema-Cache is developed by Talis and relies entirely on the Talis platform [17], which is an efficient RDF Storage and manipulation platform provided as an online service, in the spirit of cloud-based applications. Indeed, by relying on the Talis platform, Schema-Cache can abstract from any physical infrastructure and from aspects such as storage and communication scale. Finally, Cupboard is a system built collaboratively (as part of the NeOn project [8]) and that includes several components for managing ontologies. At the core of the system is the Watson Semantic Web search engine, used to index all the ontologies provided by the users, to implement the search mechanisms and to act as a base for the provided services and APIs. Cupboard also integrates the Alignment Server to create and manipulate alignments. It relies on the Oyster peer2peer ontology metadata registry [53] as a key component for manipulating ontology metadata in OMV, and on the TS-ORS [46] ontology review and rating system for user evaluation. OntoSearch2 relies on a specially designed query engine and reasoner developed by the same team: TrOWL (Tractable reasoning infrastructure for OWL 2 [59]). BioPortal relies in part on the previous developments from the same team including the Protégé ontology editor [3] and the LexGrid framework for biomedical terminologies and ontologies [54]. TONES relies essentially the OWL API [2] for managing ontologies in the OWL language, and on the OWLSight [11] ontology visualisation tool. OLS uses the OBO API [1] for the OBO language.

The mechanisms for ontology storage are influenced both by the features that each system provides and by the technical characteristics of these systems. Several systems rely on generic storage mechanisms such as databases (OLS, BioPortal) or the local file system (Cupboard, ODP, oeGov, ONKI). Other systems (OntoSearch2, Schema-Cache, TONES) use storage mechanisms specially designed for RDF data (triple stores) or for OWL ontologies (OWL API). Finally, whenever the system supports such features as advanced search, ontology browsing, representation of ontology metadata, or ontology mappings, it must deploy additional storage facilities

Library	**Components**	**Storage**	**Architecture**
BioPortal	Protégé, LexGrid	Databases; indexes; OMV metadata	Single server REST-Based communication
Cupboard	Watson, Alignment Server, Oyster, TS-ORS	Local file; indexes	Distributed components REST-Based communication
OBO Foundry	Sourceforge	CVS repository	CVS-based
oeGov	Wordpress	Local files, links to originals	Single server
OLS	OBO API	Database	Single-server
ODP	MediaWiki	Local File	Single Server
OntoSelect	Ad-hoc	Links to original ontologies	Single server
OntoSearch2	TrOWL	Triple store	Single server
ONKI	Ad-hoc	Local files; indexes	Single server
Schema-Cache	Talis Platform	Triple Store	Cloud-based
TONES	OWL API, OWLSight	OWL API	REST-based

Table A.1
Technological infrastructures underlying ontology libraries. The second column lists the key components of the technology infrastructure. The third column describes the storage mechanism. The final column describes the architecture, and specifically whether the library supports distributed architecture (and the type of communication, if yes) or whether it has a single-server model.

(e.g., indexes, metadata registries).

In terms of deployment architecture, many of the ontology library simply rely on a single Web server to deliver their functionalities. Some of the systems rely on simple Web services (in a RESTful manner [40]) to distribute and interface their components (Cupboard, TONES, BioPortal). The OBO Foundry appears here as an exception as it relies on the CVS-based architecture of Sourceforge.

A.3 Integration With The Semantic Web Infrastructure

We can consider ontologies in an ontology library to be part of the greater Semantic Web. There are several aspects to this relationship between an ontology library (even a Web-based one) and the collection of ontologies that are available on the Web.

Providing "web space" for ontologies: Many users (particularly domain experts) develop their ontologies locally. They may use an ontology library to give their ontology a Web presence. In this case, the ontology library must provide unique URIs for the ontology, and, preferably, to specific resources–classes, properties, individuals, axioms–in that ontology. In the case where an ontology library maintains versions of an ontology, a version of such URIs may resolve to the latest version (along with a version-specific URI). Some libraries, on the contrary, require that the ontology is already accessible through a URI in order to be included in a library.

Providing machine-readable URIs for ontology components: In addition to links to HTML pages that are designed for browsing, one can also have pages for ontology components that are designed for consumption by Semantic Web agents. For instance, an HTML page for a class may have RDFa embedded in it or there may be a separate URI that brings up the RDF version of the same class. The W3C outlines several best practices for publishing vocabularies [24], which differ in their choices of hash vs slash approach, and RDF vs RDFa. We distinguish ontology libraries by which of these schemes they support.

BioPortal provides URI access to ontology and pages that visualize specific ontology classes and their details. It mints its own URIs and can also generate purls that will resolve to the corresponding BioPortal pages. ODP can point to patterns elsewhere on the Web or have the users deposit patterns as OWL files on the site. ONKI stores ontologies locally and provides a URL. The OBO Foundry provides a URL for downloading the ontology. oeGov, OntoSelect, OntoSearch2, Schema-Cache link to the ontologies by their URLs; thus the ontologies must have independent Web presence to be included in the library. OLS does not provide URI-based access to ontologies. At the time of adding an ontology to the library, Cupboard gives the choice to the user either to use its original URI, or to create a new URI within the Cupboard space. This functionality is very useful in the cases where Cupboard is used as the primary publication platform for the ontology, so that entities' URIs in the published ontology can resolve into physical URIs in the Cupboard system. In addition, Cupboard employs basic *Content Negotiation* mechanisms, with the ontology URI showing either an HTML description of the ontology (metadata) or providing its RDF representation depending on the request. BioPortal uses a different URI for the HTML page of a class and for its RDF page.

A.4 Formats

At a technical level, the representation formats that an ontology library accepts is a crucial element to be considered. Here we discuss the format of the ontology itself and the languages to which it can be exported using the tools available in the library.

Supported ontology formats can include standard Semantic Web representation languages, such as RDF(S) and OWL, and more specialized ontology representation formalisms such as OBO, a format common for representing biomedical ontologies and terminologies, and RRF, a format in which the US National Library of Medicine distributes the terminologies in the Unified Medical Language System (UMLS). Some libraries can also support only a subset of a given language, such as OWL-DL.

Export can be an important additional feature of an ontology library. Indeed, while most of the libraries in our survey allow users to download an ontology only in its original format, some provide transformation facilities allowing to obtain the same ontology translated in another ontology language.

Table A.2 summarizes the ontology formats that the libraries support. For the libraries that provide transformation capabilities, the last column in the table includes the target ontology languages. Ontologies that use Semantic Web standards for representation are widely accepted by most libraries with some accepting any RDF-based representation, while others limiting to OWL or OWL-DL (TONES). The OBO Format is commonly used for representing biomedical ontologies and so the three libraries that focus on this domain (BioPortal, OBO Foundry, and OLS) accept ontologies in this format.

For the export format, Cupboard and OntoSearch2 provide a SPARQL endpoint access to their content. The OBO Foundry provides a transformation from the OBO format to OWL. Schema-Cache has a large variety of converters, while most of the others only provide the ontologies in their original format.

A.5 Features of the Library

Finally, we discuss the features that various libraries provide that ensure the value to the users—both human users accessing the collection through their Web browsers, and computer programs. We discuss the following groups of salient features:

Reasoning support: Reasoning is an essential component of using many OWL ontologies. A number of libraries provide reasoning support or contain metadata that describes the level of inference that was used to create an inferred version of an ontology (Section A.5.1).

User management: Any library that enables users to contribute content must have some level of user management, including such aspects as the ability to manage access control, provide different spaces for different users, and possibly different roles (Section A.5.2)

Notification and syndication: Some libraries enable their users to subscribe to receive notifications of changes in the library's collection, or in the ontologies themselves (Section A.5.3).

A.5.1 Reasoning

Libraries may provide mechanisms to invoke reasoning engines on ontologies in the collection. Users may request reasoning on demand, or the library software may invoke some reasoning procedures when a new ontology is loaded. For instance, one can imagine that any time a new OWL-DL ontology is loaded, the loader calls a classifier to provide an inferred version of the ontology. The library maintainers must decide what to do with ontologies that do not classify (e.g., take them down or have a flag that shows that the ontology does not classify). At this point, there is also significant variability in classifiers in terms of the features that they support. Inference information may be part of ontology metadata that the library provides. Ontology authors may be able to indicate in this metadata which inference engine they used to verify or classify their ontology. Also, for DL ontologies, the authors might want to submit both versions, an asserted version and an inferred version (e.g. some ontology authors in BioPortal take this approach [2]). The TONES repository performs classification on the fly, when the user starts ontology browsing. The repository calls the Pellet reasoner and uses OWLSight for ontology browsing.

A.5.2 User Management

Any library that enables users to contribute content must have some level of user management. Users can have different roles. In a number of libraries in our survey, only library maintainers contribute content and users do not. Thus, for these libraries user management is not an issue: The OBO Foundry, oeGov, OLS, OntoSearch2, and TONES. OntoSelect allows users to upload an ontology, but does not require them to create an account. BioPortal, Cupboard, and ODP all require users to create accounts in order to submit an ontology. There are two levels of users

[2] `http://bioportal.bioontology.org/ontologies/virtual/1083`

Library	**Accepted formats**	**Exports to formats**
BioPortal	OWL, RDF, OBO, RRF	
Cupboard	Any RDF-based, RDF/XML	SPARQL interface
OBO Foundry	OBO, OWL	OWL
oeGov	OWL (RDF, N3)	
OLS	OBO	
ODP	OWL	
OntoSelect	Any RDF-based	
OntoSearch2	OWL-DL	SPARQL interface
ONKI	RDF	
Schema-Cache	Any RDF-based, RDF/XML	RDF/XML, RSS 1.0, turtle, JSON, SPARQL interface
TONES	OWL-DL, OWL2	RDF/XML, OWL/XML

Table A.2
The ontology formats that the libraries support. The second column lists the type of file formats that the library can accept successfully. The last column lists the type of file formats that the library exports to. A blank cell in the last column indicates that the ontology is available only in its original format.

in ODP: reviews by the members of the "quality committee" are presented separately from the reviews by other users. Other libraries do not distinguish between different types of users.

Access control: In most ontology libraries that we surveyed, all users have access to all ontologies in the library. However, one can imagine the need for restricted access to some ontologies, perhaps during certain stages of their development. For instance, a group may want to make an ontology available to a limited set of users, who need to log in in order to access it, at the stage of collecting feedback from the collaborators. Ontology may become publicly available later. Similarly, there is significant interest in having libraries of proprietary ontologies, and we are sure such libraries must exist, but we did not have access to them when writing this paper. Of the libraries in our survey, however, only Cupboard provides varying level of access to ontologies, based on the notion of *ontology spaces*. Indeed, depending on whether the user is logged in and *owns* a particular ontology space, different features are activated in Cupboard (e.g., the ability to create mappings between ontologies is only available to the owner of the ontology space which contains the ontologies).

A.5.3 Notification and Syndication

Few of the libraries that we studied currently provide active notifications. In BioPortal users can subscribe to RSS feeds for all changes in one ontology (new notes or mappings, new versions), a set of ontologies, or all ontologies. Users can also subscribe to the RSS feed of the underlying blog system in oeGov. The OBO Foundry project uses the bug tracking mechanism that comes with the CVS software that they use to manage the ontologies and ontology versions. The bug tracking software provides notification mechanisms, where users can subscribe to receive emails about bugs (feature and term requests) in specific projects.