

Journal of Information Science

<http://jjs.sagepub.com/>

On the evaluation of thesaurus tools compatible with the Semantic Web

M. Mercedes Martínez-González and María-Luisa Alvite-Díez

Journal of Information Science 2014 40: 711 originally published online 14 August 2014

DOI: 10.1177/0165551514545603

The online version of this article can be found at:

<http://jjs.sagepub.com/content/40/6/711>

Published by:



<http://www.sagepublications.com>

On behalf of:



[Chartered Institute of Library and Information Professionals](#)

Additional services and information for *Journal of Information Science* can be found at:

Email Alerts: <http://jjs.sagepub.com/cgi/alerts>

Subscriptions: <http://jjs.sagepub.com/subscriptions>

Reprints: <http://www.sagepub.com/journalsReprints.nav>

Permissions: <http://www.sagepub.com/journalsPermissions.nav>

>> [Version of Record](#) - Dec 5, 2014

[OnlineFirst Version of Record](#) - Aug 14, 2014

[What is This?](#)

On the evaluation of thesaurus tools compatible with the Semantic Web

Journal of Information Science
2014, Vol. 40(6) 711–722
© The Author(s) 2014
Reprints and permissions:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/0165551514545603
jis.sagepub.com



M. Mercedes Martínez-González

University of Valladolid, Spain

María-Luisa Alvite-Díez

University of León, Spain

Abstract

Thesauri are valuable knowledge organization systems (KOSs) that support advanced information retrieval. The Semantic Web has brought a renewed interest in thesauri as a support for semantic searches and other added-value services. Tools that manage thesauri permit them to be created, edited and queried. The integrity restrictions on thesauri should also be controlled by these tools. However, there is also the possibility of thesaurus exchange, which becomes relevant in the Semantic Web context, where information exchange is a crucial facility. In fact, interoperability at the information level has received an important boost with the stabilization of the SKOS standard as a World Wide Web Consortium (W3C) Recommendation. Furthermore, the feasibility of integrating software is a valuable feature for software developers. An evaluation framework for thesaurus tools is proposed, which includes the issues of functionalities, construct support, integrity, information interoperability and feasibility of integrating software. It is original in focusing on Semantic Web conformity and interoperability.

Keywords

Knowledge representation; methodology; Semantic Web; SKOS; software evaluation; thesauri; thesaurus tools

1. Introduction

Thesauri are structured controlled vocabularies designed to facilitate information retrieval. They collect and organize terms from a domain. The ISO 2788:1986 standard [1] defines a thesaurus from two perspectives. In terms of its function, a thesaurus is a terminological control tool in which the language used in documents and by users is set out as a document language. In terms of its structure, a thesaurus is a dynamic controlled vocabulary of terms related semantically and hierarchically, applied to a given domain of knowledge. Thesauri add to information systems an ability to offer a navigational service through the thesaurus's categories or to retrieve documents indexed under one of its categories. Yet more important, however, in the context of the Semantic Web era in which we are immersed, is their value as means of sharing structured data and knowledge. In addition, they can be used as a support for concept-based searches, that is, searches in which the user looks for concepts instead of terms that appear directly in a document's content. The advanced ability of thesauri to manage synonyms, near synonyms and the subsumption of concepts makes of them ideal tools to support these searches. Examples of well-known thesauri are the Agrovoc thesaurus,¹ created and maintained by the Food and Agriculture Organization of the United Nations (FAO), the Eurovoc thesaurus² (published by the Publications Office of the European Union), the Art and Architecture Thesaurus³ (published by the J. Paul Getty Trust), the UNESCO Thesaurus⁴ and the GEMET Thesaurus.⁵

Despite the fact that thesauri have been known for a long time, the Semantic Web initiative has revitalized them as significant tools in information systems. The Semantic Web requires data that software agents can understand. To achieve this, standards that facilitate information representation in interoperable formats are needed. One such standard is

Corresponding author:

M. Mercedes Martínez-González, University of Valladolid, Valladolid, Spain.

Email: mercedes@infor.uva.es

ISO 25964-1:2011 *Information and Documentation – Thesauri and Interoperability with other Vocabularies – Part 1: Thesauri for Information Retrieval*⁶ [2]. In March 2013 a second part was published – *Part 2: Interoperability with Other Vocabularies* [3]. On the other hand, the efforts of the W3C in producing the SKOS (Simple Knowledge Organization System) standard [4], which provides a way to represent knowledge organization systems using RDF, have led, even before its stabilization, to several initiatives intended to represent various thesauri with RDF [5–8] or initiatives to manage thesauri using SKOS [9–11]. SKOS and RDF are the two Semantic Web standards that will permit thesauri to achieve a successful place in information interoperability, which is so important in web information systems.

SKOS has implied the predominance of the *concept-based thesaurus* approach over the *term-based thesaurus* approach. In term-based thesauri the vocabulary consists of a set of *terms* (words or sequences of words) related semantically and hierarchically. This is in contrast to concept-based thesauri, which contain *concepts* (abstract ideas) which are labelled with preferred labels and non-preferred labels [12]. In concept-based thesauri the relationships are established between concepts.

While thesauri were traditionally created manually or with text editors, new thesaurus tools permit them to be not only created, but also edited and used for various other tasks, such as annotating [13]. However, not all of them are made for the same purpose, permit the same operations or support integration and interoperability to the same degree. In this paper, a framework for the comparison of thesaurus tools is proposed. The term *framework* is understood here in accordance with Kitchenham [14] and Weiss [15] as a software evaluation method suitable for system comparison. It includes the methodology, the set of issues to be evaluated and the set of data (thesauri in this case) to be used in the evaluation. The set of issues includes purpose and functionality, but also other issues, which are crucial for the ability of the tools considered to be integrated into other information systems or for interoperability at the information level (the ability to exchange thesauri within different information systems) or for information integration (the ability to combine thesauri from different sources). This framework was applied to the comparison of a set of tools. These were free tools, or tools that provided a free evaluation copy, which facilitated replication by third parties.

This proposal was motivated by the need for an assessment framework that permitted selection of a tool to be used when building third-party applications that used thesauri, and that at the same time offered an editor version that could be used to undertake occasional editing of the thesauri used. Evaluations of thesaurus tools (focused on the perspective of thesaurus editors) were found to be available, as well as benchmarks for other Semantic Web tools, such as RDF stores, ontology editors or ontology APIs. However, there was no proposal that complied with our interests: evaluating and comparing free thesaurus tools, taking into account the novelties introduced by the Semantic Web on this field.

The main goal of this study was to propose a framework designed for the systematic evaluation of thesaurus tools. The issues considered in this evaluation of tools took into account the functionalities usually required in such a type of tool, with special focus on the Semantic Web perspective. This framework is not intended to be a definitive guide to evaluation of a thesaurus tool. The set of features assessed is conditioned by the needs that motivated the work. Different needs could logically imply including more, or different, issues in the evaluation. On the other hand, the target community comprises editors of thesauri and/or third-party application developers. The main goal is to assist in the *selection* of a tool; it is not to be used by tool developers to improve their tools. This also conditions the approach followed.

Other surveys do exist, made by querying thesaurus users (institutional organisations and enterprises) about their needs [16], or that result from studies about thesaurus management or comparison [17, 18]. In these surveys, features like the creation and management of thesauri, and features related to software output (display of thesauri on the screen or printer, for instance), were touched upon, but the approach presented here is original by reason of its inclusion of integration issues and the importance given to information interoperability. In addition, the consideration given to the Semantic Web standard SKOS, which stabilized as a W3C Recommendation in August 2009, is original. It was taken into account whether the SKOS Recommendation [4], or its previous Working Drafts [19, 20], were supported.

The evaluation framework is presented in Section 2 and its application to the evaluation of a set of thesaurus tools is commented upon in Section 3. The conclusions in Section 4 close the paper.

2. Evaluation framework

This section presents the framework. It includes the methodology followed (Section 2.1), its goal and cost (Section 2.2), the issues to be evaluated (Section 2.3) and the thesauri used in the tests (Section 2.4). The methodology followed in defining the framework and the evaluation was inspired by the methods put forward by Moya Martínez and Gil Leiva for thesaurus software [17], and by García-Castro and Gómez-Pérez [21] in the context of the SEALS project. In García-Castro and Gómez-Pérez [21] a general methodology for benchmarking in the Semantic Web was proposed. The methods adopted in this work took some ideas from this general proposal. However, the target users and the main goal are different, and require a different methodology. In this case, the goal is to select a tool, and the perspective taken is that of a

user, while the general methodology proposed by García-Castro and Gómez-Pérez was designed to assist developers in the improvement of the tools assessed. These differences impose different requirements. While tool improvement requires a high degree of exhaustiveness in tests in order to detect any weaknesses that should be tuned, tool users are not normally inclined to use comprehensive tests in selecting a tool, as these may become overwhelming because of their extent. In addition, other works consisting of surveys of thesaurus users [16] or the evaluation of various functionalities [18] were taken into account as complementary guidance.

As noted in the Introduction, this study is original in its inclusion of interoperability issues and the special attention given to compatibility with Semantic Web standards (RDF/SKOS). However, other important issues, such as functionalities, should not be forgotten. The functionality of a thesaurus tool determines whether it is of interest to end users, and thus this is an important issue to review when selecting one of these tools. Additionally, the thesaurus constructs supported have a direct influence on the interoperability of a thesaurus tool: the wider the set of constructs supported, the wider will be the set of thesauri supported. Syntactic interoperability is directly influenced by the formats supported. In addition, an ability to support standard vocabularies, such as SKOS, brings a tool closer to semantic interoperability at the information level (even if achieving complete semantic interoperability may be a more complex issue). It is worth recalling here the comments by Pastor Sánchez [22] in respect of the change that the new ISO standard 25964 for thesauri implies: in this new standard the *term-based* approach is replaced by a *data model*, which brings this new standard much closer to SKOS than its predecessors.

Finally, a thesaurus tool is useful for an end user, such as a thesaurus editor or creator. However, developers of thesaurus applications have another user profile, as they search for components (in the most generic meaning of this term) that they can re-use when developing their applications. A classical way of doing this is to re-use a software package during the development process. Software integration is achieved by means of software packages that can be used by other tools (.jar packages for Java applications, widgets and others that can be integrated into web pages, services that exchange XML messages, and the like). When an API is available, it provides the interoperability needed to integrate this software into other applications. The API should be well documented, with well-defined and documented method interfaces that can be easily re-used from third-party applications [23]. On the other hand, in the internet context, web services have stabilized as an appreciated technique for the re-use of external functionality. Web services push interoperability by means of XML message exchange. In the current proposal, the presence of both options, API and web services, is checked.

2.1. Methodology

The definition of the framework included the following tasks:

- (1) Definition of the main objective, benefits, scope (type of tools) and cost of the evaluation. In addition, the issues to be evaluated and the criteria used to assess them were fixed.
- (2) Definition of sets of tests that would be used to evaluate the issues. For each issue a set of tests was designed, and they were organized into suites accordingly. For each suite, the criteria used to assess the results were also fixed.
- (3) Deciding the thesauri that would be used in tests; their characteristics were defined in accordance with the issues evaluated.
- (4) Optimization of tests: in some cases, a single successful test would be equivalent to obtaining successful results in several tests from a suite at the same time; for example, if a thesaurus with several construct types was to be successfully imported, this would be equivalent to successful import of individual thesauri with one construct type in each of them. In case of negative results, the individual tests would be performed.

The subsequent stages were:

- (5) Execution.
- (6) Analysis of results and recommendations.

2.2. Goal, benefits, scope and cost

2.2.1. Goal. The main goal was to evaluate the adequacy of thesaurus tools from the perspective of a user needing to select a tool useful for:

- (1) Editing thesauri, whether for using, changing, or building them. Furthermore, it should be possible to export thesauri for use with other tools, and to import them.
- (2) Building external applications, that use thesauri, on top of it. This may be available through APIs, web services or by other means. The requirement is that the facilities offered by the tool should allow developers of such applications to abstract from thesauri implementation details.

This framework should, in fact, be helpful as a guide for anybody needing to make such a selection. In consequence, the communities concerned with this proposal are first and foremost thesaurus users and editors, together with third-party tool developers. Furthermore, tool developers might also derive some suggestions for improving their tools, although it should be remembered that, in contrast with other proposals made in the Semantic Web context [21], this is not the main goal of this evaluation framework.

2.2.2. Benefits. The benefits expected were:

- (1) to obtain mechanisms, specifically for evaluating thesaurus tools from the perspective of their users, that could be replicated in the future;
- (2) to produce recommendations on the issues evaluated for potential users of these tools;
- (3) to acquire a deeper understanding of the state of affairs relating to thesaurus tools at the current moment;
- (4) to identify issues that need further improvements in thesaurus tools, the most striking difficulties for their users and potential sources of further work.

2.2.3. Scope. To limit the number of tools evaluated, the set selected for these experiments had to comply with certain requirements. First, they had to be as recent as, or more recent than, SKOS. Thus, only tools that provided a version later than 2004 would be evaluated, this being the date when SKOS was consolidated as the standard upon which efforts in thesaurus representation in the Semantic Web context would be concentrated. The reason for taking this date as a reference was closely linked to the decision to prioritize standard formats that facilitate interoperability related to the Semantic Web. SKOS has been by far the most widely accepted norm for thesaurus representation ever since the W3C proposed it as a standard. Naturally, only tools later than SKOS could be candidates to support it.

Second, they had to be tools created for thesaurus management, or at least there should be some previously reported experience of using them with thesauri. This requirement excluded general purpose ontology editors (Protégé, Top Braid Suite, SWOOP, etc.), dictionary-oriented tools (the Wintertree Thesaurus Engine,⁷ whose name first suggested it should be considered, is in fact a tool oriented towards systems that comply with a much more general definition of a thesaurus – *A book of synonyms, often including related and contrasting words and antonyms*⁸ – than the one adopted for this study), database management systems with support for semantic technologies or thesauri (for instance, Oracle supports a limited set of SKOS), and any other tools that could be used in any way for developing thesaurus applications, but which were not specifically designed for this purpose. Furthermore, thesaurus tools that did not include functionalities permitting the editing and modification of thesauri were excluded.

A further condition was incorporated to facilitate replication of the application of the framework: the tools had to be available at no cost or have a free evaluation version. Tools that were not available in a public repository were not considered. This final decision was taken to be coherent with the goal of being useful for thesaurus users searching for free tools, as it was intended to ensure access to the tools. This implied that such users could reproduce the tests under the same conditions as in this work.

2.2.4. Cost. The evaluation was manual. Hence, its main cost was the time used for:

- (1) planning and decision-making – this took a considerable time, because several issues had to be evaluated;
- (2) defining tests;
- (3) obtaining the instruments (thesauri, software) necessary for tests;
- (4) experimenting;
- (5) analysing the results.

It is worth noting that the evaluation of a range of issues logically increases the time needed, as compared with other evaluations focused on only one issue [24, 25]. However, it offers a more general perspective, more useful for selecting a tool than an evaluation of just one or two issues would be.

2.3. Issues evaluated

The issues evaluated are the following:

- (1) *Purpose*: the purpose for which the tool was created. This affects its functionalities. Most of the tools evaluated in Section 3 were created to facilitate the task of thesaurus users (who search in thesauri, but do not edit them) and thesaurus editors (who create and edit thesauri).
- (2) *System requirements*: system requirements condition the possibilities of using a software package. This is of particular concern in the case of users with no special aptitudes for installing and managing sophisticated software, or for users who simply want to minimize the amount of software installed on their machines.
- (3) *Functionalities*: each tool offers its own set of functionalities. The set selected included functionalities that either are common in thesaurus tools known by the authors, or which were considered relevant in previous analyses of what thesaurus tools should offer [16, 17], or emerged from the needs that motivated the definition of this framework. Thus, the final set considered is:
 - *Creating and editing thesauri*. This includes:
 - creating and deleting thesauri;
 - adding metadata that describe a thesaurus;
 - creating, deleting and editing thesaurus elements (terms or concepts, relationships, domains or concept schemes);
 - revision of thesaurus elements, that is, the possibility of marking terms or concepts as candidates for insertion, removal, modification or other treatment.
 - *Search and retrieval in thesauri*. The possibilities were classified according to the issue under consideration: (a) focus on the type of string search supported; and (b) focus on the ability to restrict the search to a given type of construct.
 - string type – here there are several possibilities: (a) exact matching; (b) strings containing the input string; and (c) strings starting with the input string;
 - construct type – this refers to the ability to restrict the searching process to certain construct types; for example, it might be of interest to search only in preferred terms, in non-preferred terms, in notes, and so forth.
 - *Navigation*. This is the ability to browse from one term to another having some relationship with it. It may be hierarchical browsing (starting at the uppermost level of a thesaurus and moving down through broader and narrower relationships), or moving from one construct to a related construct by following the relationship that links them.
 - *Merging thesauri*.
 - *Importing and exporting thesauri* to and from files.
 - Providing *historical logs* reporting modifications made to thesauri.
- (4) *Thesaurus constructs supported*: this refers to the set of basic constructs, which vary depending on whether the tool under evaluation works with term-based thesauri or with concept-based thesauri:
 - *Domains* and *Subdomains* in term-based thesauri or *Concept Schemes* in concept-based thesauri;
 - *Terms* – Preferred or Descriptor, Non-Preferred or Non-Descriptor or *Concepts*.

In addition to these, there are relationships:

- *Equivalence relationships*: UF (Used For), for Non-Preferred Terms, and USE (Uses), for Preferred Terms. In concept-based thesauri these relationships are embedded in the role assigned to each label associated with a concept (PrefLabels, NonPrefLabels).
- *Hierarchical relationships*: BT (Broader Term), NT (Narrower Term), Top Term. In concept-based thesauri these relate to the concepts, Broader Concept, Narrower Concept and Top Concept. In some thesauri, there is also a multiple hierarchy or polyhierarchy.

- *Associative relationships*: RT (Related Term), which is mirrored to a Related To relationship between concepts.
- *Notes*: Scope notes, Historical notes, Editor notes, Usage notes, etc.

(5) *Integrity management*: the integrity of a thesaurus is considered an important function of thesaurus tools by experts in thesaurus creation [12, 17]. It relates to maintaining the semantic coherence of the constructs and relationships created in a thesaurus, so that any updates that would result in integrity violation are automatically rejected by the tools. In addition, a tool that can cope with automatic integrity management should be able to insert automatically any relationships that can be derived from those inserted by end users (for example, a BT relationship implies an NT relationship).

When tools automatically check integrity, users are greatly helped in correcting errors committed while creating a thesaurus, so they can be more confident about the work done. As happens with ontologies [26], this is especially relevant when thesauri are large. This behaviour could also be expected in thesaurus software used in third-party applications. If it can handle thesaurus integrity (for instance through the corresponding exceptions when a violation occurs), the applications that use it are freed from the task of checking integrity. Thesaurus standards, such as ISO 2788, or SKOS, establish certain integrity conditions that a thesaurus, or a KOS represented with SKOS, should always respect. In term-based thesauri, integrity conditions determine what is, or is not, valid in a thesaurus. The integrity conditions for thesauri are:

- *Uniqueness*: the concepts modelled are unique, that is, there cannot be duplicated elements in a thesaurus. This means that there is only one Preferred Term (or Preferred Label) for each concept; that there are no two equal Domains, that is, a domain cannot be repeated; that Preferred Terms and Domains are disjoint, that is, a Preferred Term cannot also be a Domain (or, in SKOS dialect, ConceptSchemes are disjoint from Concepts); and that Preferred Terms, Non-Preferred Terms and Lexical Variants are disjoint sets (PrefLabels, AltLabels and HiddenLabels are pairwise disjoint properties).
- Only Preferred Terms can participate in semantic relationships. This means, for instance, that a Non-Preferred Term is not allowed to participate in these relationships. In concept-based thesauri this means that only concepts participate in relationships.
- Some relationships are incompatible: two terms cannot be related simultaneously by two relationships of these incompatible types. Therefore, BT and NT relationships are not compatible with an RT relationship; in other words, two terms already related by a BT or NT relationship cannot also be related terms (RT relationship).
- Cycles involving hierarchical relationships are forbidden. For example, if *A* is broader than *B*, *B* cannot be broader than *A*. This also holds when both hierarchical and associative relationships are involved in the cycle. Furthermore, a top term cannot be narrower than other terms in the hierarchy.
- Some relationships require the existence of an inverse relationship. For example, if *A* NT *B*, then *B* BT *A*.
- When a term is deleted, all the relationships in which it participates should also be deleted.

(6) *Information interoperability*: the ability to import or export thesauri represented in different formats is crucial for the aim of interoperability. Moreover, a capacity to support SKOS representations and to exchange components of a model is of importance for the sake of Semantic Web goals. It should be possible to know at this level which formats, with their syntactical variants, are supported for import and export. The focus is on Semantic Web standards, RDF/SKOS, with syntactic variations, RDF/XML, N3 and others. SKOS-XL, proprietary schemes, or even ISO 25964 could be considered at this level. However, at the moment when this evaluation framework was prepared, ISO 25964 was not yet supported by the tools evaluated.

(7) *Software interoperability and integration*: software can be made re-usable by means of APIs offering a set of libraries provided with interfaces describing class definitions and behaviour – in other words, methods that permit users to interact with it. The APIs of interest are generic APIs, designed for use with any thesaurus.

Another possibility is the use of web services. A web service is a ‘software system designed to support interoperable machine-to-machine interaction over a network’ [27]. Such services permit applications to be converted into web applications. Following the GEMET thesaurus documentation,⁹ which includes an API for accessing thesauri through web services, the first web service API that existed for thesauri was the US National Agricultural Library Thesaurus API. Around 2001, the RDF SKOS specification arrived, and with it, the SKOS web service API. However, it was abandoned around 2004. According to the GEMET documentation, one possible reason could be that it is very SOAP-specific and that it is monolingual.

2.4. Thesauri for evaluation

An ad-hoc thesaurus was designed for testing the *create and edit* functionalities. This thesaurus contains the minimal components needed to check construct support in thesaurus tools. Two additional thesauri were also used in order to check import capabilities with external thesauri (i.e. those not created with the tool being evaluated). As the formats of these thesauri were not produced with any of the tools evaluated, they offered a better check on the ability of these tools to support external formats, that is, their capacity to inter-operate with thesauri produced externally.

- (1) The *Ad-hoc Thesaurus* is used to test the creation of a new thesaurus. A mini-thesaurus was built with 1 Domain (subject field), 3 Subdomains (micro-thesauri), 13 Preferred Terms or Descriptors, 1 Non-Preferred Terms or Non-Descriptors and 1 Scope Note. This thesaurus is a simplification, inspired by the Eurovoc thesaurus, with the minimal elements necessary to facilitate tests relating to construct support and integrity. It is the thesaurus proposed to build with the edit functionalities of the tools evaluated. It contains the constructs included in the tests: Domains (subject fields), Subdomains, Descriptors, Non-Descriptors, Synonyms, Notes, lexical variants, BT/NT relationships, RT relationships and polyhierarchy. It includes two levels of hierarchical relationship (BT1, BT2, NT1 and NT2). It is shown in Figure 1. *Geography* is the subject field. The three microthesauri are preceded by the key 'MT' in the figure. Top Terms are recognized by the 'TT' keyword. A simpler version is used when polyhierarchy or several domain levels are not supported: this minimal version contains just two microthesauri, *Europe*, *Regions of EU Member States*, which eliminates the subject field (*Geography*) and the polyhierarchy (*Spain* only appears once as narrower term, instead of twice).
- (2) The *UKAT Thesaurus*: SKOS/RDF representation.¹⁰ UKAT took the UNESCO Thesaurus as its starting point, incorporating terms from other structured vocabularies (such as *Library of Congress Subject Headings*). It is organized into seven fields of knowledge (Education, Science, Culture, etc.), each of which is divided into microthesauri. Overall, the UKAT thesaurus has 83 microthesauri (such as Educational policy, Educational administration, etc.). Each microthesaurus is a grouping of terms. The number of descriptors (preferred terms) in the thesaurus is 13,976, and there are 6638 non-preferred terms. There are NT/BT and RT relationships, and notes. UKAT is poly-hierarchical, although the majority of terms have only a single broader term. Polyhierarchical terms have several broader terms and belong to several microthesauri. It is publicly available for download in SKOS-Core format (compliant with the SKOS-Core 1.0 RDF Schema), in English. Its file size is 9.9 MB. In this representation, there are Dublin Core attributes to describe the thesaurus. Micro-thesauri are represented with ConceptSchemes. Fields of knowledge are not explicitly represented.
- (3) A version of the *Eurovoc Thesaurus* represented with SKOS. This representation of Eurovoc was obtained by transforming the Eurovoc XML files obtained from the Publications Office of the European Union through the application of a set of transformations. The result conforms to the SKOS Recommendation. Its file size is 4.8 MB. Eurovoc is a thesaurus maintained by the EU Publications Office. It is organized into 21 fields of knowledge, subdivided into 127 micro-thesauri. There are 6645 descriptors, of which 519 are top terms and 7756 non-descriptors. Some terms from fields 72 (*Geography*) and 76 (*International Organizations*) are poly-hierarchical. The thesaurus, terms, notes and relationships were mapped according to the guidelines presented in the Appendix of the SKOS primer document [28]. Microthesauri and fields of knowledge were represented with ConceptSchemes.

3. Application

The framework proposed in the previous sections was applied to a set of tools. The evaluation issues were organized into a set of test suites, which were applied to these tools. The results of each test were normalized to a range of possible result values, obtained by adapting the proposal in García-Castro and Gómez-Pérez [21] to the actual problem of thesaurus tool evaluation. The core set of values is *yes/no*, which reflects if the issue is *supported* and if the result is semantically correct (i.e. correctly *modelled*). In Tables 1–3 both criteria are condensed into a single value for the sake of brevity, so that, if the value shown in a cell is *yes*, it means that the issue is supported by the tool and the result is semantically correct.

3.1. Selection of tools

The tools selected were found by searching on web pages that maintain lists of thesaurus tools, Semantic web technologies or SKOS tools. They include two collections of tools for managing taxonomies or thesauri, a list of Semantic Web

Geography
MT Europe
TT Southern Europe
NT1 Spain
RT regions of Spain
UF Kingdom of Spain
TT Eastern Europe
NT1 Bulgaria
MT Regions of EU Member States
TT Regions of Bulgaria
NT1 Severen tsentralen (Bulgaria)
NT2 Gabrovo region
NT2 Ruse region
MT Political geography
TT NATO countries
NT1 Spain

Figure 1. Ad-hoc thesaurus for testing the *create* and *edit* functionality.

Table 1. Functionalities.

	MultiTes	One-2-One	PoolParty	SKOSEd	TemaTres	ThManager
Creating and editing	Yes	Yes	Yes	Yes	Yes	Yes
Searching and retrieving	Yes	Yes	Yes	Yes	Yes	Yes
Navigation	Yes	Yes	Yes	Yes	Yes	Yes
Merging	No	Yes	Yes (from SKOS files)	Yes (from Protégé Ontology import)	No	No
Importing and exporting	Yes	Yes	Yes	Yes	Yes	Yes
Logs	No	Yes	No	No	No	No

Table 2. Constructs supported.

	MultiTes	One-2-One	PoolParty	SKOSEd	TemaTres	ThManager
Basic constructs	Thesauri, Terms	Thesauri, Terms	ConceptScheme, Concept	ConceptScheme, Concept	Thesauri, Terms	Thesauri, Terms
PrefTerms/NonPrefTerms	Yes	Yes	Yes	Yes	Yes	Yes
Equivalence relationships	Yes	Yes	Yes	Yes	Yes	Yes
Hierarchical relationships	Yes	Yes	Yes	Yes	Yes	Yes
Associative relationships	Yes	Yes	Yes	Yes	Yes	Yes
Notes	Yes (scope, private)	Yes (scope)	Yes (SKOS-supported)	Yes (SKOS-supported)	Yes (scope, historical, catalogue, private)	Yes (scope, definition)

development tools found in one of the two wikis maintained by the Semantic Web Deployment Working Group,¹¹ and one page in Wikipedia dedicated to SKOS. The tools selected for evaluation were:

- (1) PoolParty 2.7;
- (2) MultiTes;
- (3) SKOSEd 1.0: SKOS editor for Protégé-OWL 4.0;
- (4) ThManager 2.0;
- (5) temaTres 1.2;
- (6) One-2-One. This tool replaces TermTree.

Table 3. Integrity, interoperability and integration.

	MultiTes	One-2-One	PoolParty	SKOSEd	TemaTres	ThManager
Automatic insertion of inverse relationships	Yes	Yes	Yes	Yes	Yes	No
Automatic deletion of relationships	Yes	Partial	Yes	Yes	Yes	No
Control of hierarchical relationships	Yes	Yes	Yes	Yes (depends on reasoner used)	No	No
Thesaurus import	Yes (.txt)	Yes (Trim, ascii, eDocs, etc.)	Yes (SKOS)	Yes (SKOS)	Yes (Tab files)	Yes (SKOS)
Thesaurus export	Yes (.txt, XML, SKOS)	Yes (Excel, CSV, XML, SKOS)	Yes (SKOS)	Yes (SKOS)	Yes (.txt, SKOS, Zthes, MADS)	Yes (SKOS)
Integration	Yes (Web services)	No	Yes (Web services)	Yes (SKOS API)	Yes (Web services)	No

Nevertheless, there are further tools that were not included in our evaluation, even though they comply with the tool selection criteria proposed in Section 2.2.3, such as ISGAT¹² or Skosify,¹³ which were not yet available when this tool selection started. As the main interest of this work is the framework that supports comparisons, and not the final results for a given set of tools, a decision was taken to restrict the set of tools assessed to the original set.

3.2. Results

An analysis of the evaluation of the six tools is presented below. The first two issues, *purpose* and *requirements*, are not commented upon individually, because there are no significant observations worthy of mention. As noted in Section 2.3, the purpose was basically editing in all cases. The requirements in some cases include a need for prior installation of some other software package, such as MySQL or Protégé, but all of the items necessary can be freely downloaded and installed in accordance with the instructions accompanying them without too much technical experience being required.

3.2.1. Functionalities. Table 1 summarizes the support of functionalities by the tools examined. Basic functionalities are supported by all of them. In addition, Dublin Core metadata are generally supported, even if not in all cases (only MultiTes and One-2-One did not support Dublin Core). The revision of thesaurus elements was enhanced in term-oriented tools such as TemaTres or One-2-One, while in concept-based tools this feature seems to be given less prominence. There were some differences in searching, although most of the tools supported string searches in several forms and all of them permitted searching for a string containing the input string. However, while some tools, such as ThManager and TemaTres, permitted searching only in Preferred Terms (or Preferred Labels), other tools extend the support to Alternative Labels, Notes or even Hidden Labels. Navigation is performed through relationships. However, One-2-One, PoolParty and ThManager also offer the possibility of browsing the thesaurus hierarchy. The merging of thesauri, which could be useful in some cases, is supported by One-2-One and by tools for which Semantic Web technologies played a major role in their development. This is logical, as merging of ontologies is an important issue for ontology developers in the Semantic Web context. Import and export of thesauri are generally supported, although there are some variations that will be commented upon later in relation to information interoperability. Historical logs seem to be treated as irrelevant for the developers of these tools, with the exception of One-2-One.

3.2.2. Thesaurus constructs. The thesaurus constructs supported by each tool are shown in Table 2. In the first row, the basic constructs indicate what type the tool is: term-oriented or concept-oriented. All the tools are able to support the constructs necessary for the majority of thesauri. An interesting difference was found in the manner in which polyhierarchy is supported. While term-oriented tools, such as TemaTres, support it by duplicating the descriptor with more than one ascendant, in concept-oriented tools such as PoolParty, the concept was never duplicated, even if it participated in two *narrower* relationships. As for subdomains, this is a feature missing in most of the tools. In addition, there are differences in the range of Notes supported.

3.2.3. Integrity. Table 3 summarizes the results obtained in respect of the issues of integrity, interoperability and integration. There are differences with regard to integrity management. PoolParty and MultiTes passed all of the tests. At the

opposite extreme, ThManager did not deal with integrity at all. In between came One-2-One, SKOSed and TemaTres, which passed some of the tests, but failed others. Integrity management is an important issue for thesaurus creators, as the responsibility for checking it may, or may not, fall completely on them. It is worth noting that the status of a tool as a thesaurus tool or a SKOS tool has implications for certain integrity conditions, specific to thesauri, which are not supported by SKOS tools. For example, cycles in hierarchical relationships are allowed in SKOS (the resulting KOS is consistent), while in thesauri if a term *A* is broader than another term *B*, then *B* cannot be broader than *A*. This could be a criterion for preferring a tool, particularly when creating big thesauri. By contrast, for simply manipulating existing thesauri, this should not be a relevant point in influencing decisions.

3.2.4. Information Interoperability. The tools that use Semantic Web technologies for thesaurus storage, such as PoolParty or SKOSed, support a broader range of Semantic Web formats (RDF, N3, Turtle, and others) for importing and exporting thesauri. Furthermore, JSON is available as an alternative format to XML for the most recent tools. This is logical, because RDF stores usually offer the possibility of importing and exporting to all of these formats. In contrast, other tools should take the responsibility for format manipulation. It was observed that very few tools stated which SKOS version they support. This could have some impact, as the SKOS Recommendation introduced changes with respect to previous Working Drafts which affect checking of integration. For example, the domain of the *skos:inScheme* property was changed from *Concept* to being 'effectively the class of all resources *rdfs:Resource*' [4], so that what is valid in one version is not valid in the other. In relation to SKOS, it is a drawback of TemaTres and One-2-One that they are not able to import SKOS files. When importing external thesauri, the probability of receiving them in SKOS format is high, and this is increasing with the spread of the SKOS standard.

3.2.5. Software integration. A general and standard API for thesauri seems to be an ideal that is not yet close. In fact, the abandoning of the SKOS web service API seems to be a bad omen in respect of the possibility of such an API being made available within a reasonable time. It seems that at present each big thesaurus or thesaurus tool develops its own API for accessing it. The GEMET API is very well documented and exhaustive, and it provides very useful ideas on what a thesaurus API should contain, and its implementation through web services. The SKOS API is another interesting effort relating to SKOS management. The TemaTres approach, based on web services, has the benefit of simplicity in encouraging the use of its web services. The web services with free access to their documentation (GEMET, TemaTres) covers querying facilities. The set of functionalities related to thesaurus creation and editing is not covered. As for proprietary tools, such as PoolParty, these could not be checked, as access to documentation for this is restricted to PoolParty clients.

4. Conclusions

This paper presents an evaluation framework for thesaurus tools. It allows systematic analysis of these tools in order to assess their support for functionalities typical of this type of tool, but it includes a novelty with respect to previous evaluation proposals for thesaurus tools: consideration of the advances that the Semantic Web technologies and standards offer for thesaurus management and sharing. As such, the framework has two valuable features. The first is that it was designed specifically for thesaurus tools, which may be of use for users searching for a guide to comparisons of this type of tool. The second is the inclusion within the evaluation issues of some that reflect the advances in thesaurus representation that have come with the Semantic Web. Even though some general guidelines for evaluating Semantic Web tools had previously been proposed, there had hitherto been no proposals specifically adapted for thesaurus tools.

The paper presents and summarizes the main components of the evaluation framework in such a way as to keep it within reasonable space limits. As a consequence, the tests are not enumerated, but they can be directly inferred from the description of issues presented in Section 2.3.

Although SKOS is one of the preferred standards for representing thesauri, and although some of the tools evaluated use SKOS as the main language for representing thesauri, it is important to note that this is not an evaluation of SKOS tools. SKOS is not intended for exclusive use with thesauri, which implies some significant differences from thesauri [4] which would have an impact on the methodology or approach followed in evaluating each type of tool.

It is of interest to note that the topic of ontology merging, which receives considerable attention in the Semantic Web community, was reflected in the tools closer to the Semantic Web, which supported thesaurus merging. The interaction of the thesaurus community with the Semantic Web community has brought to thesaurus tools certain functionalities that previously had only a secondary role. On the other hand, noting the virtual absence of some functionalities, such as historical notes, it may be worthwhile reconsidering their inclusion in future revisions of the framework.

One of the more relevant conclusions is a realization that the distinction between the more traditional set of thesaurus tools and Semantic Web-oriented tools is crucial. Tools that support Semantic Web standards are in an advantageous position for thesaurus exchange, as reflected in the broader support they offer for importing and exporting thesauri. This is logical, as it is well known that the use of standards is beneficial for information exchange.

However, it is also possible to emphasize within this second set of tools the distinction between those designed for thesaurus management and those designed for KOS represented with SKOS. The first group uses SKOS as a basic element in thesaurus representation, but has functionalities (and thus, a user interface) adapted for thesaurus users. They also include several integrity restrictions specific to thesauri that are not included in the SKOS recommendation. An example of this first group is PoolParty. The second group, being SKOS focused, does not adapt functionalities to the specificities of thesauri and, logically, does not include the integrity restrictions that are not considered in the SKOS recommendation, which means that thesaurus-specific restrictions are not checked by these tools. For users accustomed to handling thesaurus tools the first group is more intuitive, while the second set might be easier for users coming from the ontology engineering community.

In respect of the capacity to integrate software it will be necessary to observe developments in these tools in the future, which should clarify whether SKOS-oriented APIs or thesaurus-oriented APIs gain more favour. There is also a need to watch the evolution of these tools in the near future so as to see if SKOS-oriented tools are able to offer thesaurus users the support they are seeking, or if it continues to be necessary to build tools specially adapted to the specificities of thesauri.

The framework presented here was designed taking into account the ISO 2788 standard, which is the one currently supported by thesaurus tools. For the future, it will be appropriate to revise certain tests after the ISO 2596 standard becomes widely supported in new versions of thesaurus tools, especially those tests related to the issue of information interoperability. A review of integrity conditions by checking this new standard against SKOS is envisaged, in order to ensure that these are kept up to date.

Notes

1. <http://aims.fao.org/website/AGROVOC-Thesaurus/sub>
2. <http://europa.eu/eurovoc>
3. http://www.getty.edu/research/conducting_research/vocabularies/aat/
4. <http://www2.ulcc.ac.uk/unesco/>
5. <http://www.eionet.europa.eu/gemet>
6. ISO 25964-1:2011 revises ISO 2788:1986.
7. <http://www.wintertree-software.com/dev/thesdb/javasdk.html>
8. Definition taken from <http://www.thefreedictionary.com/>
9. <http://taskman.eionet.europa.eu/project/zope/wiki/GEMETWebServiceAPI>
10. Available at <http://www.ukat.org.uk/> (last visited: 2013/07/03)
11. <http://www.w3.org/2006/07/SWD/>
12. <http://www.um.es/isgat>
13. <http://code.google.com/p/skosify/>

Funding

This research received no specific grant from any funding agency in the public, commercial or not-for-profit sectors.

References

- [1] ISO 2788:1986 *Documentation – Guidelines for the Establishment and Development of Monolingual Thesauri*. International Organization for Standardization (ISO), 1986.
- [2] ISO 25964-1:2011 *Information and Documentation – Thesauri and Interoperability with other Vocabularies – Part 1: Thesauri for Information Retrieval*, 2011.
- [3] ISO 25964-1:2011 *Information and Documentation – Thesauri and Interoperability with other Vocabularies – Part 2: Interoperability with Other Vocabularies*, 2013.
- [4] Miles A and Bechhofer S. SKOS Simple Knowledge Organization System reference. W3C Recommendation 18 August 2009, <http://www.w3.org/TR/2009/PR-skos-reference-20090818/> (2009, accessed September 2013).
- [5] Lacasta J, Nogueras-Iso J, López-Pellicer F, et al. ThManager: An open source tool for creating and visualizing SKOS. *Information Technology and Libraries* 2007; 26: 39–51.

- [6] Faro S, Francesconi E, Marinai E, et al. EUROVOC Studies LOT2 D2.3. Report on execution and results of the interoperability tests. Technical Report 10118. Florence: Publications Office of the EC, Institute of Legal Information Theory and Techniques ITTIG, 2008, http://eurovoc.europa.eu/drupal/sites/all/files/Presentation-D2.3_Mapping_results.pdf (accessed September 2013).
- [7] Polo Paredes L, Alvarez Rodríguez JM and Rubiera Azcona E. Promoting government controlled vocabularies for the Semantic Web: The EUROVOC thesaurus and the CPV product classification system. In: *Semantic interoperability in the European digital library: Proceedings of the first international workshop, SIEDL 2008*, Tenerife, 2 June 2008. Lecture Notes in Computer Science, Vol. 50212. Berlin: Springer, 2008, pp. 111–112.
- [8] Smedt JD. SKOS Extensions for the EUROVOC thesaurus. In: *Third annual European semantic technology conferences (ESTC)*, Vienna, 2–3 December, 2009.
- [9] Gerbe O and Keerherve B. A Model-driven approach to SKOS implementation. In: *Internet and Web applications and services (ICIW)*. IEEE Computer Society, 2010, pp. 484–488.
- [10] Koller A. SKOS thesaurus management based on linked data with PoolParty. In: *Third annual European semantic technology conference*, Vienna, 2–3 December 2009, http://triplify.org/files/challenge_2009/poolparty.pdf (accessed September 2013).
- [11] Mayr P, Zapilko B and Sure Y. Establishing a multi-thesauri-scenario based on SKOS and cross-concordances. In: *Proceedings of the 2010 international conference on Dublin Core and metadata applications*, Pittsburg, PA, 2010, pp. 194–196, <http://dcpa.pers.dublincore.org/pubs/article/view/1031> (accessed September 2013).
- [12] Pastor Sánchez JA, Martínez Méndez FJ and Rodríguez-Muñoz JV. Advantages of thesaurus representation using the Simple Knowledge Organization System (SKOS) compared with proposed alternatives. *Information Research* 14(4), <http://informa.tionr.net/ir/14-4/paper422.html> (2009, accessed August 2013).
- [13] Hildebrand M, Van Ossenbruggen J, Hardman L, et al. Supporting subject matter annotation using heterogeneous thesauri: A user study in Web Data reuse. *International Journal of Human–Computer Studies* 2009; 67(10): 887–902.
- [14] Kitchenham B, Pfleeger S and Fenton N. Towards a framework for software measurement validation. *IEEE Transactions on Software Engineering* 1995; 21: 929–944.
- [15] Weiss A R. Dhrystone Benchmark: History, analysis, scores and recommendations. White paper, EEMBC Certification Laboratories, 2002.
- [16] Severino F. *The term development in the thesauri of international organizations*. London: Frank Cass, 2007.
- [17] Moya Martínez G and Gil Leiva I. Evaluación de software de gestión de tesauros. *Ciencias de la Información* 2001; 32(3): 3–23.
- [18] Willpower Information, Software for building and editing thesauri, <http://www.willpowerinfo.co.uk/thesoft.htm> (2012, accessed September 2013).
- [19] Miles A and Bechhofer S. SKOS Simple Knowledge Organization System reference. W3C Working Draft 25 January 2008, <http://www.w3.org/TR/2008/WD-skos-reference-20080125/> (2008, accessed September 2013).
- [20] Miles A and Bechhofer S. SKOS Simple Knowledge Organization System reference. W3C Working Draft 9 June 2008, <http://www.w3.org/TR/2008/WD-skos-reference-20080609/> (2008, accessed September 2013).
- [21] García-Castro R and Gómez-Pérez A. Benchmarking in the Semantic Web. In: Cardoso J and Lytras M (eds) *Semantic web engineering in the knowledge society*. Information Science Reference IGI Global, 2009, pp. 341–370, <http://taskman.eionet.europa.eu/projects/zope/wiki/RESTInterface> (accessed September 2013).
- [22] Pastor-Sánchez JA. ISO-THES: Ampliando SKOS a partir de la norma de tesauros ISO 25964. *Anuario ThinkEPI* 2013; 7: 189–193.
- [23] Bloch JJ. How to design a good API and why it matters. In: Tarr PL and Cook WR (eds) *OOPSLA companion*. New York: ACM Press, 2006, pp. 506–507.
- [24] Garcia-Castro R and Gómez-Pérez A. Guidelines for benchmarking the performance of ontology management APIs. In: *Proceedings of the 4th international Semantic Web conference (ISWC 2005)*. Lecture Notes in Computer Science, Vol. 3729. Berlin: Springer, 2005, pp. 277–292, http://oa.upm.es/4866/1/ISWC_2005.pdf (accessed June 2013).
- [25] García-Castro R and Gómez-Pérez A. Benchmarking the RDF(S) interoperability of ontology tools. In: *Proceedings of the 19th international conference on software engineering and knowledge engineering (SEKE 2007)*, Boston, MA, 2007, http://oa.upm.es/4212/1/SEKE_2007.pdf (accessed June 2013).
- [26] Noy NF, Sintek M, Decker S, Crubezy M, Ferguson RW and Musen MA. Creating Semantic Web contents with Protégé-2000. *IEEE Intelligent Systems* 2001; 16 (2): 60–71.
- [27] Haas H and Brown A. Web services glossary. W3C Working Group, Note 11 February 2004, <http://www.w3.org/TR/ws-gloss/> (2004, accessed September 2013).
- [28] Isaac A and Summers E. SKOS Simple Knowledge Organization System primer. W3C Working Group, Note 18 August 2009, <http://www.w3.org/TR/skos-primer/> (accessed September 2013).