Design and Implementation of an Open Engine System for Science and Technology Knowledge Organization Systems

Authors: Wang Ying, Zhixiong Zhang, Li Chuanxi, Liu Yi, Tang Yijie, Zhou Zijian, Qian Li, Fu Honghu, Wang Ying

Date: 2016-01-25T00:00:00+00:00

Abstract

[Purpose] To achieve the sharing and utilization of the Scientific and Technological Knowledge Organization System (STKOS). [Application Background] Constructing knowledge organization abstract: [Purpose] To achieve the sharing and utilization of the Scientific and Technological Knowledge Organization System (STKOS). [Application Background] Constructing an engine system for the organic storage and access of knowledge organization systems is a prerequisite for the effective utilization of knowledge organization systems. [Method] Building a semantic storage and indexing system, a semantic query and inference kernel, and an STKOS API that support the retrieval, browsing, association, and navigation of various STKOS elements, and providing open query and inference interfaces to external parties. [Result] This engine system supports the construction of STKOS publishing service platforms and the application of STKOS in third-party retrieval service systems. [Conclusion] Through the STKOS open engine system, scientific literature information institutions and researchers in China can conveniently and effectively utilize STKOS.

Full Text

The Design and Implementation of Open Engine System for Scientific & Technological Knowledge Organization Systems

Wang Ying[1], Zhang Zhixiong[1], Li Chuanxi[2], Liu Yi[3], Tang Yijie[3], Zhou Zijian[3], Qian Li[1], , Fu Honghu[1]

[1](National Science Library, Chinese Academy of Sciences, Beijing 100190, China)

[2](China Great Wall Asset Management Corporation, Beijing 100045, China)
[3](Wuhan Library, Chinese Academy of Sciences, Wuhan 430071, China)
(University of Chinese Academy of Sciences, Beijing 100049, China)

Abstract

[**Objective**] To enable the sharing and utilization of the Scientific & Technological Knowledge Organization System (STKOS). [**Context**] Building an engine system for organic storage and access of knowledge organization systems is a prerequisite for their effective utilization. [**Methods**] We constructed a semantic storage and indexing system, a semantic query and reasoning kernel, and STKOS APIs that support retrieval, browsing, association, and navigation of various STKOS elements, and provided open query and reasoning interfaces for external use. [**Results**] The engine system supports the construction of the STKOS publishing service platform and the application of STKOS in third-party retrieval service systems. [**Conclusions**] Through the STKOS open engine system, scientific and technological literature information institutions and researchers can conveniently and effectively utilize STKOS.

Keywords: Knowledge organization; Engine system; Query interface

2 Related Research

To achieve effective utilization of knowledge organization systems, numerous institutions have conducted research and developed systems for publishing and sharing knowledge organization systems, supporting user and machine access through online retrieval and browsing services, APIs, and Web services.

The UMLS Terminology Services provide online retrieval and browsing interfaces as well as remote access Web service APIs. Each complete UMLS download version includes the MetamorphoSys tool for local installation and customization of the Metathesaurus. UMLS Metathesaurus data is stored in RRF or ORF files, both following an Entity-Relationship (ER) model, enabling joint queries across multiple files through a series of ER-like "query graphs"[2]. The MetamorphoSys tool improves retrieval efficiency for its RRF browser by building index files [3]. Additionally, MetamorphoSys can create SQL scripts to load customized UMLS Metathesaurus and Semantic Network data into relational databases such as Oracle and MySQL, facilitating further UMLS utilization [3].

The NCI Enterprise Vocabulary Services (EVS) provides integrated retrieval and browsing services for the NCI Thesaurus, NCI Metathesaurus, and other terminologies through the open-source LexEVS terminology server [4]. LexEVS is built on the LexGrid model, enabling formal representation and storage of multiple vocabulary formats including OBO format, NCI Thesaurus OWL language, and UMLS RRF format. It utilizes Java APIs and distributed LexBIG APIs for querying and accessing LexGrid model vocabulary data, creates Lucene

indexes to enhance query performance, and provides both SOAP and REST service interfaces [5].

The OCLC Terminology Services offers retrieval and browsing for vocabularies such as FAST, GSAFD, LCSH, and MeSH, supporting multiple data formats including HTML, MARC XML, SKOS, and Zthes, as well as SRU CQL syntax-based retrieval. Through Web service interfaces, it provides XML, RDF, and JSON-encoded outputs to third parties, supporting both REST and SRU/SRW access methods [6].

The AIMS website provides online search, browsing, and download services for the AGROVOC multilingual agricultural thesaurus. The AGROVOC Web service uses SOAP technology to access RDF-version AGROVOC data. AIMS' s VocBench editing tool enables web-based open collaborative editing of AGROVOC. The edited and maintained AGROVOC adopts the SKOS-XL conceptual framework and has been published as Linked Open Data (LOD), connecting with other multilingual knowledge organization systems in agriculture. Its linked data version uses the AllegroGraph database for storage, provides a machine-accessible SPARQL endpoint, and generates human-readable HTML pages through the Pubby tool [7].

Domestically, the Institute of Scientific and Technical Information of China has built the Chinese Scientific and Technical Vocabulary System, which provides services through the vocgrid platform[1], allowing users to access complete knowledge of vocabulary entries [8]. The system uses MySQL database for storage, with API interfaces divided into data access and business logic layers, employing Web service technology to encapsulate and publish application service interfaces for various system modules [9].

After in-depth investigation of foreign terminology registration and data service systems, Ou Shiyan [10] proposed recommendations for developing China' s terminology registration and service system: adopt emerging semantic web and linked data technologies to build integrated systems with both terminology registration and service functions, provide human and machine access to registered vocabulary metadata and content, employ RESTful Web services to build terminology services, and support vocabulary content publication as linked data.

From the above research, we observe that knowledge organization systems are gradually transitioning to data formats such as SKOS/RDF and OWL, with storage mechanisms shifting from relational databases to RDF semantic repositories, and external access methods expanding from SOAP-based Web services to include REST and SPARQL endpoint access. Therefore, considering STKOS' s specific requirements, this project developed the STKOS open engine system using advanced semantic repository, indexing, query, reasoning, and interface technologies.

---

3 System Design and Implementation

3.1 Overall Framework

The STKOS open engine system is an open knowledge organization engine developed based on the Scientific & Technological Knowledge Organization System (STKOS) content. According to STKOS data characteristics, it constructs a high-performance, reliable knowledge storage and indexing system and an STKOS retrieval query and semantic reasoning kernel engine, supporting the construction of the STKOS publishing service platform and providing externally callable STKOS API interfaces. Through standard open query and reasoning service interface protocols, it serves domestic institutions and third-party systems.

The overall framework of the STKOS open engine system, shown in Figure 1 [Figure 1: see original paper], consists of four layers: storage and indexing layer, query and reasoning function layer, STKOS API layer, and open query and reasoning interface layer.

(1) **Storage and Indexing Layer:** Virtuoso serves as the underlying RDF database, storing all RDF triples of the STKOS super thesaurus and ontologies. Based on this, RDF indexes and Solr multidimensional indexes are built to support queries, reasoning, and applications based on the knowledge storage and indexing system.

(2) **Query and Reasoning Function Layer:** Built upon the storage and indexing layer, this layer further implements SPARQL query optimization and performs corresponding index scheduling and analysis processing. It employs a combination of pre-indexing reasoning and real-time reasoning during retrieval to implement semantic reasoning for STKOS.

(3) **STKOS API Layer:** According to the metadata and structural characteristics of the STKOS super thesaurus and ontologies, this layer designs and implements STKOS API interfaces to support upper-level access and applications. The STKOS API layer comprises three sub-layers: STKOS Basic API, STKOS Upper API, and STKOS Solr API, providing basic data access and application-oriented API interfaces for STKOS data.

(4) **Open Query and Reasoning Interface Layer:** Through an interface manager, this layer controls the external services of STKOS open query and reasoning interfaces. Based on a Web service architecture, it encapsulates STKOS engine functions using both SOAP protocol and REST protocol Web services, and provides a SPARQL endpoint to support external direct SPARQL queries on RDF data.

3.2 Data Storage

The Scientific & Technological Knowledge Organization System (STKOS) consists of a controlled vocabulary system and an ontology library. The STKOS

super thesaurus is expected to contain no fewer than 5 million scientific and technical terms, 800,000 standardized concept names, 60,000 domain ontology concepts, and 200,000 properties [1]. To support STKOS openness and linking, both the STKOS super thesaurus and ontologies are stored and utilized as RDF triples. Based on thorough investigation of semantic repositories [11], the high-performance Virtuoso database was selected as the RDF repository to achieve effective storage of STKOS super thesaurus and domain ontologies at ultra-large scale, supporting query, reasoning, and application.

Virtuoso is a universal RDF database where all triples together with named graph information are stored in the quad table RDF_QUAD, with four columns (G, S, P, O) representing Graph, Subject, Predicate, and Object, respectively. To store the STKOS super thesaurus and domain ontologies, named graphs are created separately. According to the STKOS super thesaurus metadata specification, data conversion rules are defined, and automatic conversion programs are written to transform the super thesaurus into RDF triples via Virtuoso' s JDBC connection and store them directly in the database, while using Virtuoso' s built-in tools to directly import domain ontology OWL files.

3.3 Index Design

To support fast retrieval and full-text search, both Virtuoso indexes and Solr indexes are designed for underlying storage and upper-layer applications. Through in-depth analysis of STKOS super thesaurus and domain ontology data characteristics, it was found that the number of RDF triple Predicates is minimal compared to Subjects and Objects. For query requirements, RDF repositories are more suitable with Predicate as the primary index item. Therefore, PSOG and POGS RDF indexing patterns are adopted to build RDF indexes on Virtuoso' s RDF_QUAD table.

Although Virtuoso database provides full-text retrieval functionality, its generality limits performance for large-scale data retrieval. Consequently, Solr' s outstanding performance and flexible organization in full-text and faceted retrieval are leveraged to build indexes for the STKOS super thesaurus and domain ontologies, providing full-text retrieval services via HTTP protocol. The STKOS Solr API establishes index fields for main super thesaurus elements such as terms, concepts, and category classes based on metadata description information and data requiring full-text retrieval. For domain ontologies, indexes are built on RDF triple sets of classes and instances to support index construction for STKOS basic elements and meet the demand for rapid full-text retrieval response.

3.4 Query Optimization and Reasoning Strategies

Based on the STKOS underlying repository and index construction, query performance is further optimized and reasoning strategies are formulated to support STKOS-oriented query and reasoning functions. Query optimization focuses on

both execution efficiency and processing procedure optimization. For Virtuoso' s SPARQL query execution efficiency, optimization strategies include leveraging Virtuoso' s built-in caching mechanism and tuning database configuration parameters such as NumberOfBuffers and MaxDirtyBuffers to improve query performance on STKOS physical storage. Additionally, different query optimization strategies are formulated for basic elements in the STKOS super thesaurus and domain ontologies. For example, for string matching retrieval requirements, Solr indexes are used to compensate for Virtuoso' s SPARQL query speed issues in fuzzy matching. For slow SPARQL COUNT queries, statistics for different STKOS element types are computed after data batch import and written into Solr indexes, rewriting COUNT queries as direct data retrieval operations.

To enable extensive use and deep mining of STKOS, semantic reasoning strategies are established. On one hand, common reasoning results are pre-computed offline based on STKOS data characteristics and physically stored or indexed. For instance, hierarchical relationships such as hypernym-hyponym relations between concepts in the super thesaurus and subclass relations in ontologies are stored as triples in physical storage. Since STKOS applications frequently use the transitive reasoning functionality of these relationships, query traversal methods are employed to store paths from top-level concepts/classes to current concepts/classes as hierarchical structures for direct query access, saving reasoning time. On the other hand, Virtuoso' s built-in backward reasoning engine and predefined SPARQL query templates support online reasoning requirements, such as transitive relations like rdfs:subClassOf, rdfs:subPropertyOf, owl:sameAs, and inverse relation reasoning.

3.5 STKOS API

To enable access to STKOS basic elements, the STKOS API is developed in Java, encapsulating Virtuoso SPARQL queries and Solr queries. Based on data processing and application levels, the API is divided into three layers: STKOS Basic API, STKOS Upper API, and STKOS Solr API.

(1) **STKOS Basic API:** For physically stored STKOS, basic data encapsulation is implemented according to various basic elements, defining corresponding fundamental interfaces. Super thesaurus basic interface definitions include: retrieving term names, synonym IDs, broader term IDs, linked concept IDs, etc.; retrieving standardized concept definitions, preferred term IDs, linked category class IDs, etc.; retrieving category class preferred names, non-preferred names, category numbers, broader category class IDs, etc.; and retrieving thesaurus titles, alternative titles, types, etc. Domain ontology basic interface definitions include: retrieving class names, parent class URIs; retrieving data property and object property names, domains, range lists, parent properties; and retrieving instance names, belonging class URIs, etc.

(2) **STKOS Upper API:** Built upon the STKOS Basic API, this layer pro-

vides interface definitions for retrieving detailed STKOS information, including: retrieving broader term names, narrower term IDs and names, synonym names, top-level terms, term extension attributes and values, and inferred broader/narrower terms; concept preferred names, non-preferred names, category class names, etc.; and category class broader/narrower class IDs and names. Domain ontology upper interface definitions include: retrieving class subclass URIs and names, hierarchically structured classes obtained through reasoning, top-level classes, specified hierarchical class lists, specified class instance URIs and names and inferred instances, and instance property values. Additionally, corresponding interfaces are defined for STKOS element type queries and string queries.

(3) **STKOS Solr API:** The STKOS Solr API further implements practical application query requirements such as fuzzy matching and similarity ranking, mainly including query interfaces for term, concept, category class, class, instance, and other related information, as well as offline information query interfaces. It implements functions such as string-based full-text retrieval, fuzzy matching, and STKOS statistical information retrieval.

3.6 Open Query and Reasoning Interface

The open query and reasoning interface implements external services for the open engine system. Based on a Web service architecture, STKOS engine functions are encapsulated using both SOAP protocol and REST protocol Web services [12]. Above the STKOS storage and indexing layer, query and reasoning function layer, and core STKOS API layer, an interface manager controls the external services of STKOS APIs [12].

Through analysis of third-party application requirements, STKOS external service interfaces are specifically divided into four functional categories implemented using the SOAP protocol: **Browsing Service Interface** for the super thesaurus category table and category classes, including methods for retrieving top-level category classes, broader/narrower category classes for specified classes, and category class hierarchical structures; **Retrieval Service Interface** for concepts and terms in the super thesaurus, encapsulating underlying Solr index query methods, including retrieving concepts and terms with different attributes such as broader, narrower, and related terms; **Association Reasoning Service Interface** implementing association query interfaces to discover relationships between concepts or terms in the super thesaurus, including retrieving related terms for specified terms, category classes to which terms belong, and semantically related terms for specified concepts; and **General Service Interface** implementing interface operation status, version change information, and statistical information for the STKOS super thesaurus (statistics on category classes, concepts, terms, etc.) [12].

Through analysis of the STKOS super thesaurus metadata specification, STKOS

interfaces are divided into three categories implemented using the RESTful protocol: **Concept Service Interface**, a collection of methods related to STKOS super thesaurus concepts including browsing, retrieval, and association reasoning; **Category Service Interface**, implementing interface methods based on STKOS super thesaurus category classes and category tables; and **General Service Interface**, similar to the general service interface in SOAP, implementing interface operation status and statistical information [12].

Additionally, a SPARQL endpoint for STKOS is implemented using Virtuoso' s built-in tools (Figure 2 [Figure 2: see original paper]), allowing users to query RDF data from different named graphs of the STKOS super thesaurus and domain ontologies through SPARQL query language.

---

4 Application

The STKOS open engine system can effectively support the construction of the STKOS publishing service platform and provide externally callable STKOS open query and reasoning service interfaces through standard interface protocols for third-party system usage.

4.1 Supporting STKOS Publishing Service Platform Construction

Currently, the STKOS open engine system has achieved effective storage of the STKOS initial version data. In the Virtuoso database, it stores approximately 540,000 concepts, 1.42 million terms, and 10,000 category classes from the STKOS super thesaurus, totaling about 72.57 million triples including names, definitions, properties, and relationships, as well as over 1,000 classes and 10,000 instances from three domain ontologies, totaling approximately 150,000 triples. To support rapid upper-layer application access to underlying data, the engine system provides 259 interfaces, with statistics shown in Table 1 .

Table 1: STKOS API Interface Statistics | Level | Interface Count | |—-| ————-| | STKOS Basic API | | | STKOS Upper API | | | STKOS Solr API | |

Based on the STKOS open engine system, the project team constructed the STKOS publishing service platform[1] (Figure 3 [Figure 3: see original paper]), designing the indexing system and system architecture according to application requirements to provide online STKOS retrieval, browsing, navigation, and display services, enabling users to conveniently acquire, consult, and utilize the scientific and technological knowledge organization system. Additionally, the platform provides STKOS customization and download functions to meet the needs of different institutions and users for domain knowledge organization systems, maximizing the sharing and reuse of STKOS achievements.

The main functions of the STKOS publishing service platform include: (1) **STKOS Retrieval Function:** Searching terms and concepts in the STKOS super thesaurus through keywords or IDs; (2) **STKOS Browsing Function:**

Establishing tree navigation for category tables to enable classified concept browsing, and providing tree browsing for domain ontologies; (3) **STKOS Customization Function:** Allowing authorized users to select concepts or terms from the STKOS super thesaurus through retrieval and browsing functions to customize and download STKOS subsets; (4) **Statistical Function:** Providing statistics on STKOS current version data.

4.2 Supporting Third-Party Retrieval Service System Applications

To support third-party systems in utilizing the STKOS super thesaurus, the engine system provides open query and reasoning interfaces based on both SOAP and REST protocols[1], with invocation addresses and usage methods shown in Figure 4 [Figure 4: see original paper]. Based on the STKOS open engine system, the project team built an STKOS-based retrieval service experimental system using the Chinese Academy of Sciences Literature Integrated Retrieval System as a foundation, accessing STKOS super thesaurus data through open query and reasoning interfaces.

The retrieval service experimental system implements intelligent retrieval functions based on the STKOS super thesaurus. Figure 5 [Figure 5: see original paper] shows the retrieval effects achieved using open query and reasoning interfaces when users input the keyword "GENE". The retrieval system uses the SOAP protocol's getStandardWord interface to search for "GENE" and returns corresponding concepts from STKOS, helping users achieve standardized term retrieval. Using the getConceptNarrowerList and getConceptBroaderList interfaces, the system retrieves narrower and broader concepts for corresponding concepts, recommending semantically related vocabulary to users and indicating hit counts for these terms in the current result set (marked as 1 in Figure 5 [Figure 5: see original paper]). Secondly, through the getConceptSemanticRelatedList interface, the system obtains semantically related terms for concepts, prompting them in the result list if they appear in retrieved documents (marked as 2 in Figure 5 [Figure 5: see original paper]). Additionally, the queryConcept interface is used to retrieve fuzzily matched related concepts, providing search recommendations and enriching original literature retrieval results (marked as 3 in Figure 5 [Figure 5: see original paper]).

The development and construction of the STKOS open engine system represent research and exploration into knowledge organization system storage and access mechanisms. The system employs advanced semantic repository, indexing, query, reasoning, and interface technologies to build an effective open knowledge organization engine that supports STKOS publishing service platform construction and STKOS-based retrieval applications, enabling public query and browsing of STKOS while supporting third-party system development and utilization of STKOS. Future work will further enhance system maturity and service capabilities, expand application scope, and establish it as an information infrastructure supporting effective STKOS utilization by domestic information institutions and research organizations.

References

[1] Sun Tan, Liu Zheng. Methodology Framework of Knowledge Organization System for Scientific & Technological Literature [J]. Library & Information, 2013(1): 2-7.

[2] UMLS Database Query Diagrams [EB/OL]. [2014-03-20]. http://www.nlm.nih.gov/research/umls/implemen

[3] MetamorphoSys Help [EB/OL]. [2014-03-20]. http://www.nlm.nih.gov/research/umls/implementation_reso

[4] NCI Enterprise Vocabulary Services [EB/OL]. [2015-03-24]. http://evs.nci.nih.gov/.

[5] LexEVS 6.x Architecture [EB/OL]. [2012-05-20]. https://wiki.nci.nih.gov/display/LexEVS/LexEVS+6.x+A

[6] Vizine-Goetz D. Terminology Services [EB/OL]. [2013-06-08]. http://tspilot.oclc.org/resources/overview.pdf

[7] Caracciolo C, Stellato A, Morshed A, et al. The AGROVOC Linked Dataset [J]. Semantic Web, 2013, 4(3): 341-348.

[8] Zhang Yunliang, Xu Shuo, Zhu Lijun, et al. Chinese Scientific and Technical Vocabulary Systems–Semantic Resource for Deep Content Analysis of S&T Information Resources [J]. Library and Information Service, 2010, 55(4): 100-105.

[9] Shi Xin, Qiao Xiaodong, Zhang Zhiping, et al. Research and Implementation of the Web Service of Chinese Technological Vocabulary System [J]. New Technology of Library and Information Service, 2008(12): 37-42.

[10] Ou Shiyan. A Review of Foreign Terminology Registries and Terminology Services [J]. Journal of Library Science in China, 2014, 40(5): 110-126.

[11] Zou Yimin, Zhang Zhixiong, Qian Li, et al. Technical Analysis and Application of Semantic Repository-Virtuoso [J]. Journal of the China Society for Scientific and Technical Information, 2013, 32(1): 13-21.

[12] Liu Yi, Tang Yijie, Zhou Zijian, et al. Research and Construct of the Service Interface in STKOS Sharing Infrastructure [J]. New Technology of Library and Information Service, 2014(7-8): 9-16.

Author Contributions

Zhang Zhixiong: Proposed research ideas, revised final manuscript;
Wang Ying: Drafted manuscript;
Wang Ying, Li Chuanxi: Designed research scheme, conducted experiments and implementation;
Li Chuanxi: Third-party retrieval service system application experiments;
Liu Yi, Tang Yijie, Zhou Zijian: Proposed external service interface scheme and implementation;
Qian Li, Fu Honghu: STKOS publishing service platform construction.

Received: 2015-04-27
Revised: 2015-05-08

---

[1]http://stkos.las.ac.cn/
[1]http://stkos.las.ac.cn/webservice/services

Note: Figure translations are in progress. See original paper for figures.