MEDINFO 2019: Health and Wellbeing e-Networks for All
L. Ohno-Machado and B. Séroussi (Eds.)

doi:10.3233/SHTI190478

TBench: A Collaborative Work Platform for Multilingual Terminology Editing and Development

Panpan Deng, Yujing Ji, Liu Shen, Junlian Li, Huiling Ren, Qing Qian, Haixia Sun

Institute of Medical Information, Chinese Academy of Medical Sciences, Beijing, China

Abstract

Terminology facilitates consistent use and semantic integration of heterogeneous, multimodal data within and across domains. This paper presents TBench (Termilology Workbench) for multilingual terminology editing and development within a distributed environment. TBench is a web-service Java tool consisting of two main functionalities that are knowledge construction (i.e.extended model based on ISO25964, batch reusing and constructing multilingual concept hierarchy and relationships) and collaborative control in order to achieve custom extensions, reuse, multilingual alignment, integration and refactoring.

Keywords:

Controlled Vocabulary, Software, Architecture

Introduction

Many terminology, taxonomy and ontology editing tools have been developed, and some are world-famous, such as Protégé [1]. However, many domain experts, even terminology experts, still think that there are no suitable tools for them to build what they want. In addition to economic factors, complexity of operations and cognitive burden may be two important reasons. The former is usually manifested as new hierarchy trees and relationship instances, which can only be created one-by-one. Other lingual terms can only be created as non-preferred terms, are unable to reuse related thesauri and ontologies flexibly, have difficult to extend attributes and relationships for knowledge units, and so on. The latter is mainly manifested as lack of understanding of the meaning of user interface function labels. Motivated by these shortcomings, TBench was designed and developed.

Methods

TBench is based on a classical three-tier architecture (Figure 1), structured through a data layer, a functional layer and an interface layer.

Data Layer Design

Data Description Format Design

According to SKOS and ISO25964 [2], TBench has a customizable data model including core and extended models for all knowledge units (versions, thesaurus, concept group, concept, term, note, etc.) respectively. Each terminology system or top concept tree can define its own extension model on term type, hierarchical and semantic relationships, descriptions of attributes, etc.

Data Storage Format and Data Format Converter Design

All data are stored in an Oracle relational database. To facilitate users conversions of data between different formats (TXT, JSON, XML and RDB) and semantic description models (SKOS, OWL and RDF), data format converters are also designed.

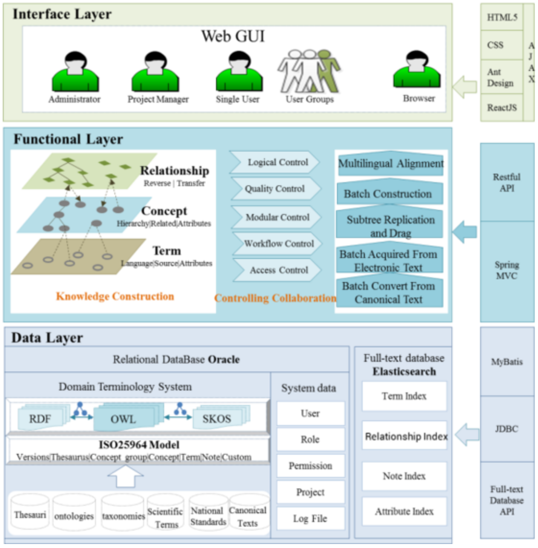

Figure 1- The Architecture of TBench (Termilology Workbench)

Functional Layer Design

The functional layer consists of two main parts: knowledge construction and collaborative control. In addition to basic editing functions, batch construction and multilingual mechanisms are also designed and developed to support cross-language knowledge construction easily and efficiently.

Batch Construction by Flexible Reuse and Operation

Multiple ways are designed for reuse and batch construction, inculding:

1. Using automatic batch conversion techniques, canonical text can be batch converted to hierarchical structure, sibling concepts, non-preferred terms and semantic relationships.

2. Through grammatical rules, concepts and relationships from electronic text are quickly acquired and integrated with a clear workflow.
3. For reuse and integration, fast replication and simple drag operations of subtrees within or between terminology systems are designed and implemented.
4. Some objects having common features can be batch created using 'indication' relationships with the same domain.

Multilingual Mechanism Design

In order to create a terminology system containing multilingual terms, each concept has a preferred term in every language. To provide the culture feature, all objects (concepts, relationships, attributes, etc.) of TBench have a 'language' description. Exact, inexact and partial equivalence mappings among preferred terms of the same concept in different languages can also be created to reveal cultural differences. For instance, the mapping between 'aircraft' and '飞机' is inexact.

Controlling Collaboration Between Users

A role-based access control (RBAC) model was adjusted to support flexible configuration between users, roles, permissions, and task state automatic transitions by using direct permissions setting and the introduction of resource work states. Locking and unlocking were used to resolve knowledge unit conflicts during real-time collaborative editing.

Interface Layer Design

Different Tasks, Different Workflows, and Different User Interfaces

To provide multi-user personalized usability, customizable user interfaces were also designed. Users can choose the function modules, data items in modules, and system labels. For instance, menus, buttons and attributes can be described in natural language depending on the user.

Results

TBench has been developed (Figure 2) in Java™. In addition to the data converter, all the above designs have been implemented. It is being used for cross-institutional constuction of the Chinese Clinical Terminology System (CSCT), consisting of 14 categories of 207,000 concepts organized hierarchically, and 65 semantic relationships.

Conclusions

TBench, a web-service tool, can be used for collaborative building and maintaining of controlled vocabularies, thesauruses, classifications, taxonomies and ontologies. It improves several key points, such as customizable data models and interfaces for different users, batch construction operations for relationships and concept hierarchies, and so on. The reliability of TBench is being tested and verified with CSCT. Crowdsourcing of construction will continue to be studied in the near future.

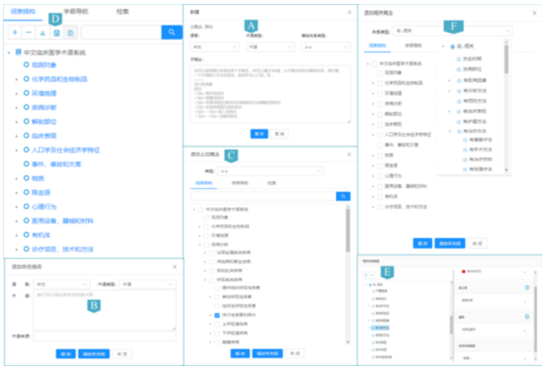

Figure 2 – TBench (Terminology Workbench) user interface. (A) Concept creation interface. It can automatically batch convert from canonical text and configuration language, term type, and relationship type. (B) Non-preferred terms batch creation interface. (C) Upper concept configuration interface. Selecting multiple upper concepts is an effective way to establish poly-hierarchical relationships. (D) Tree structure interface enables users to create, delete, copy, cut, paste and drag a concept or subtrees. (E) Relationship configuration interface enables users to customize various new relationships with their domain and range. (F) Relationship instances with the same domain can be batch created by choosing multiple concepts as a range.

Acknowledgements

This work was supported by the National Key Research and Development Program of China (grant #2016YFC0901901), Medical and Health Technology Innovation Program of The Chinese Academy of Medical Sciences (grant #2017-I2M-3-014).

References

[1] Musen MA. The Protégé Project: A Look Back and a Look Forward. *AI Matters*. 2015, 1(4):4–12.

[2] International Standards Organization (ISO). ISO 25964 - 1:2011 Information and documentation -- Thesauri and interoperability with other vocabularies -- Part 1: Thesauri for information retrieval, 2011.

[3] Abburu S, Babu G S. Survey on Ontology Construction Tools[J]. *International Journal of Scientific & Engineering Research, IV* (2013), 1748-1752

Address for Correspondence

Corresponding author : Haixia Sun
Email: sun.haixia@imicams.ac.cn