

An Object Oriented Knowledge Management Platform

Zheng WU¹

SiQi WU

E-mail: bill@pentedge.com

wilson@pentedge.com

Pentedge Software and Consultancy Co. Ltd., China

Abstract. LOGIST is an object oriented knowledge management platform. Presented in this paper are some innovative features that support highly efficient bilingual ad hoc query operation that lays solid ground to an open platform for knowledge management for wide area of domain specific applications. Exemplified with some publicly available linked data, we will show you how an ad hoc query may be constructed, modified, translated, derived before sent to heterogeneous data sources to get solution and, in particular, how “Hot Menu” technology, probably one of the most profound highlights that are worthy your attention, is used to increase usability. Simply Push down Ctrl key before drag and drop a sentence, e.g. a query, to a menu item; the sentence is then transformed into a hyperlink as a new menu item ready for your option. With ad hoc query, “Hot Menu” technology well paraphrases the concept of Dynamic Menu and will help you through collecting data across the Web or accumulating them in a local database. This way a knowledge base may be created incrementally totally tailored to your own needs and maybe also shared with other peers across Web.

Keywords. Knowledge Management, Knowledge Reengineering, Computational Natural Language, Machine Translation, Semantic Web, Ad Hoc Query, User Interface.

¹ Corresponding Author.

Introduction

With an objective to create a legal information system, the development of project Logist² started 10 years ago. Issues to be addressed include domain specific conceptual object modeling, reasoning and information retrieval to achieve a public knowledge management platform facilitating many aspects of legal professionals' day to day operation, including legislation, enforcement, and education.

1. Consideration

As costs, maintenance, wideband, burden of computing load are many issues fall within systematic consideration, what we have in our mind is a distributed formalized conceptual network system with a really-user-friendly-interface based on open-world-assumption.

In order to allow end users more roles in an open knowledge management system, a platform with a "really-user-friendly-interface" is considered inevitable. No direct exposure of underlining technical details, e.g. XML, RDB etc., to end users are among essential prerequisites. Preferably it should be a natural language text based talking-machine or question-answering system that works very much the same way as a lawyer does.

As the easy access is simply something that prompts widespread involvement of end users, the easy access to Web pages eventually leads internet to an era of World Wide Web. We anticipate "really-user-friendly-interface" will play the same role in leading internet to the time of Web of Data, Semantic Web or Web of Law.

2. Innovative Features of Logist: Toward a Semantic Web Operating System

Although legal information system is our ultimate development target, Logist is designed as a common knowledge management platform system. In particular, as a participant of the Semantic Web Challenge 2014, we will show you,

- a) conceptual object system are constructed, manipulated in an incremental manner, i.e. knowledge accumulation, through either ad hoc queries or object tree browser and creation wizard;
- b) Ad hoc queries are constructed, modified, translated, derived either based on or from an existing query or a query template, before sent to heterogeneous data sources to get meaningful solution.
- c) In addition to contextual editor and translation wizard, copy& paste, drag& drop and other mouse operation are intensively used in modifying sentence component and transforming one sentence into different sentence patterns or different languages, e.g. English, Chinese or machine code.

² <http://www.pentedge.com>.

- d) Hot menu is used to increase usability of Semantic Web application. In addition to contextual editor, quick sentence editing and distributed data model structure etc., hot menu is amongst all that make Logist a handy platform suggesting a new way of collaborating with Semantic Web from a knowledge management perspective.

2.1. The Introduction of Formalized Natural language

Natural language hides the complexity of code syntax while prompting complex semantic content. Based on HTML hyper text, Logist document performs well in presenting RDF statements, SPARQL, rules and even complex sentences to end users, as natural language sentences are used in ask, describe, construct, update, select, entailment rule and derive type of queries and rules, which hides URI and other syntax details but will ultimately be transformed into machine readable code, e.g. SPARQL and Jena rule.

As free natural language text usually leads to ambiguities, in Logist, only a limited set of sentence patterns are allowed. This has been doing well in disambiguation of word sense to get quality transformation from natural language into machine readable code. Not only for disambiguation but also for fast editing, it is the formalized sentence patterns with which Logist becomes an efficient platform supportive to advanced fast editing, making conceptual object defining, entailment rule creating, query constructing, etc. a process much more convenient than ever before.

Putting limits on the number of sentence patterns might be its downside of formalized natural language as your "freedom" is restricted to some extent, but improved performance, efficiency and effectiveness might be its bright side either.

Fortunately, with our sentence pattern management system, the number of sentence patterns is not dead locked one; In Logist, sentence patterns are treated just like other conceptual objects; sentence pattern data is stored in RDF document for persistence; sentence pattern data is also created, shared and installed in an incremental manner just like other linked data. Although manipulation of some of sentence patterns demands a little bit training and essential linguist knowledge, most of them may be carried out by ordinary users.

We are anticipating an open platform system which shares sentence pattern data just like legacy operating systems or web browsers share font, alphabet or dictionary data across the world. With our sentence pattern management system, complex sentences, e.g. verb sentences (sentence contains a verb), sentences with attribute clause and compound sentences, are expected to be heavily used to increase conciseness and comprehensiveness to make it a platform approaching to natural language based operating system.

2.2. Object System Browsing and Construction

Logist works very much the same fashion as an operating system that interfaces user machine interaction with respect to knowledge management.

2.2.1. Special feature

Hierarchical tree browser: classes, subclass and instances of class are displayed in tree browser.

Property sheet: properties and values of each object appear in a dedicated property sheet upon receiving focus, e.g. when mouse click upon an tree node or hypertext in representing the object.

Contextual Popup Menu: right click on one object and choose from Contextual Popup Menu to get access to options right for this object for manipulation or browsing.

Contextual Wizard for Creation of Object: right click on one object and choose create subclass, for example, from the popup menu; the contextual wizard will guide you through the process creating a new object instance, e.g. subclass, instance of class, etc.

Quick restructuring: drag and drop one object into another to restructure the hierarchical class and instance relationship by automatically adding a statement into designated data model, e.g. establishing value for a property, e.g. RDFS:subClassOf or RDF:type .

2.2.2. Special Design Consideration and Lesson Learned

Essentiality: Beside natural language interface, an object browsing system is essential to an intuitive user interface to help navigating through the object network which is especially useful to a user often get lost in the complicated network system.

Tree is concise: Tree is used in representing the hierarchical object system to make the structure intuitive and better understandable, but special table control is required to avoid duplicated branch of tree.

Topology network is another option: as such topology network, especially one that centered on only one object, might be better solution and thereof is to be considered in future release of Logist.

2.3. Fast sentence construction and derivation

Logist works very much the same way as MS-Excel or HTML page except that Logist has many wizards for semantic content editing and contains sentences that are executable.

2.3.1. Special features

Contextual focus system: click on one position of a sentence or sentence block, system will decide contextually which sentence component or sentence to receive focus and to receive further operation, e.g. editing, translation, etc..

Quick clone of sentence and sentence component: drag and drop, copy and paste to move or clone a sentence, a block of sentence or a sentence component where you like them to be.

Contextual editor: with contextual editor, a sentence constituent component may be modified abiding to predefined semantic or syntax rules, e.g. constraint value searching based on name input, date time format control, on fly input construction, etc..

Translation wizard: with translation wizard, a sentence, include ad hoc query, rules and other complex sentences, may be translated into different sentence patterns or in different languages.

Ad hoc query based on sentence editing: drag drop or copy paste one sentence or block into the query body to quickly put constraint on query solution. The sentence may be further modified to fit the query task.

On fly Ad hoc query construction: conversion of sentence or block of sentences into a query by a mouse click combined with CTL, ALT or SHIFT key is another way for quick ad hoc query construction. Whatever comes up in your mind on whatever and wherever you see a sentence or block of sentences, you may make a query on the fly, modify it, send it to a designated data source and see the result.

Remove and override through update: while update operation is used to add a property value(s) to one object, remove and override options are used to allow one particular property value(s) to be removed, or added but all other value(s) to be removed from of the very property of the very conceptual object.

2.3.2. Special Considerations and Lessons

The range constraint on each sentence component input: as drag drop, copy paste operations are frequently used in modification, validation must be made against the range predefined for the concerned component; illegal input is rejected or highlighted; to the highest extent avoiding illegal semantic and syntax input.

Sentence management system: sentence management system is inevitable so that creation of useful sentence pattern data can be easily achieved by professionals as well as ordinary users, e.g. high school students, linguists, domain experts, etc. Sentence pattern data should be used just like font, alphabet and dictionary data.

Derivation generator: the derivation generator of Logist is proved useful to generate simple derivation of one sentence, e.g. queries, rules, equivalent or similar sentence patterns, e.g. tense and time of verb sentence.

Ad hoc is all about extensibility and flexibility: Neither "do everything button" nor "factory made menu" will work all the time, all the place and for all the people, as such ad hoc query will be something that is desperate in need particularly in a system which is based on open-world-assumption, e.g. Semantic Web, Web of data, Linked data etc.

Collaborative knowledge management: in combination with ad hoc query and system feedbacks, e.g. solution of queries, heterogeneous data from different sources, e.g. RDF file, SPARQL End point, RSS feed, etc. regardless their owners' whereabouts, may be easily collected and accumulated to or reused through a local data model, i.e. local database or knowledge base, totally tailored to end user's needs, in a collaborating and incremental manner, simulating to the process of leaning and knowledge accumulation by human beings.

2.4. Data Source as an Object

Correspondent to RDF model concept, data source object, instance of OWL:Ontology or its subclass, is created facilitating consuming of heterogeneous data, for end users to have easy access to RDF or owl file, SPARQL endpoint, etc.

2.4.1. Special Features

On fly data source creation: Choose from the prompted popup menu when right click on Ontology class object or its subclasses, a contextual wizard will guide you

through the creation of a new instance of data source, e.g. RDF document, SPARQL end point, RSS feed, etc.

Import data: choose Import from the popup menu when right click on a data source object, a data model for the data source will be created and added to Logist main data model as a sub model. Upon request it may also be removed from the main model.

Receive ad hoc queries: Each query has a data source input component so that specific data source may be chosen to receive the query in focus.

Editable data source input component: drag drop, copy paste one data source object to a query's data source input component to change it with a new replacement; or double click on the component to edit it through the contextual editor prompted; you may also do a constraint search by the name of data source or by the predefined range; for an input not existed in the system, you may choose to create a new one right away by clicking on the button laid on right up corner of the contextual editor.

2.4.2. Special Considerations

Scalability of the system: flexible data model structure makes the overall system scale totally based on end users' needs.

Properties of data source object: each instance of data source class will have different properties and values; modifications are supposed to be a guided procedure by a wizard to ensure proper function unless carried out by an advanced user.

Reuse of data source object: data source object is created and reused just like other conceptual object. Especially a local data path property is attached to get a local copy of data sources for efficiency as wideband is always a practical issue in concern.

2.5. Hot Menu: Profound Technology that Worthy Your Attention



Figure 1. The First Hot Menu You will see from Logist.

Figure 1 is a screen shot of a sample hot menu and probably the first menu you will see from Logist as you start the application.

2.5.1. *Special Features*

Easy access to customized functions: click on one of the items listed in the menu, e.g. Describe, the evaluation result of a query, e.g. describe concept, will display in the feedback box underneath the menu body, except it is an update type query, which will always show query content for your confirmation before sent to backend system.

Customization and confirmation: hold down ALT key while clicking on an item, only the query content, e.g. describe concept, will display in the feedback box for your confirmation or modification before sent to the system.

Showing in a new tab page: hold down CTL key while clicking on an item, the query content, e.g. describe concept, will display in a new tab page.

Creating new item: push down CTRL key before drag and drop a sentence, especially a query with some of your modification, to a menu item; the sentence is then transformed into a hyperlink as a new menu item immediately ready for your option.

Rename: you should be able to rename each of the items through a prompted contextual editor by a double clicking on it.

Reuse or share: Just like any other sentence, the customized menu is then maybe saved into a Logist document for reuse or share across the web just like a HTML file.

2.5.2. *Special Considerations and Lessons*

“Hot Menu” is about usability: while all other features are something about “X-ibilities”, e.g. flexibility, feasibility, etc. the hot menu technology is about usability which well paraphrases the concept of Dynamic Menu. This is somewhat explained by the fact that not all people like the tricky of kitchen work; some only want food; some is ok with a little bit food combination; while others like to be professional chiefs. For users of different professions levels, hot menu is just the technology to have all needs addressed in a perfect manner.

Game changing technology is inevitable: hot menu is somewhat a game changing technology in the sense that a menu item is no long something that heavily relied on IT developer to create or provide maintenance. In other word, hot menus and the likes are desperate in need to get balanced burdens among IT professionals and end users. We doubt whether it is normal if an industry have been always so busy in creating or learning different advanced programming languages to meet endless user interface requirements, rather than put most of their energy in data itself, e.g. creation, mining, sharing and reuse.

“Hot Menu” is not everything: “Hot Menu”, ad hoc query, etc are useful for non-time-critical application, e.g. web search, legal information system, diagnosis expert system, eLearning system and so on. In combination with web browser, web search engine, word processing program, spreadsheet application and the likes, semantic web applications will make the legacy applications even “wiser” as a result of improved information processing automation and quality of data.

Sentence is Object: Reifying a sentence into an object means that URI is also assigned to each sentence as well as its constituent component so that manipulation of sentence as an object becomes possible. Although reifying a RDF statement is not new as it is often used in Jena model operation, reifying a complex sentence or its derivations is very useful in the creation of another complex sentence, e.g. ad hoc queries, sentence with attribute clause, etc.

3. Appendix

The following is the evaluation result against to each of the criteria predefined by the Semantic Web Challenge 2014.

3.1. Minimal requirements

1. The application has to be an end-user application, i.e. an application that provides a practical value to general Web users or, if this is not the case, at least to domain experts. It should show-case functionalities that the use of semantic web technologies can bring to an application: **Yes, Logist is an end-user application which provides very practical value to Web users.**
2. The information sources used
 - should be under diverse ownership or control: **Our data sources are from around the world under diverse ownership and control;**
 - should be heterogeneous (syntactically, structurally, and semantically) : **Our data sources include all RDF file, RSS feed, SPARQL END POINT, etc. and**
 - should contain substantial quantities of real world data (i.e. not toy examples): **data from <http://www.270a.info/>, e.g. [Eurostat - Linked Data](#) and RSS feed are all sample of real world data.**
3. The meaning of data has to play a central role. **The introduction of formalized natural language is all about the meaning of data.**
 - Meaning must be represented using Semantic Web technologies: **Yes, Semantic Web technologies are used throughout;**
 - Data must be manipulated/processed in interesting ways to derive useful information: **ad hoc query, hot menu item, translation, etc. are all interesting and innovative way to manipulated/processed data;** and
 - this semantic information processing has to play a central role in achieving things that alternative technologies cannot do as well, or at all: **to our knowledge, they are new and have not seen in other applications so far;**

3.2. Additional Desirable Features

“In addition to the above minimum requirements, we note other desirable features that will be used as criteria to evaluate submissions.”

1. The application provides an attractive and functional Web interface (for human users): **Logist is Web interface for human users either as high school students or domain experts; hot menu may be attractive and functional to ordinary users while ad hoc query system may be attractive and functional to advanced users.**
2. The application should be scalable (in terms of the amount of data used and in terms of distributed components working together). Ideally, the application should use all data that is currently published on the Semantic Web: **As Jena**

grounds the running of Logist, all data that is currently published on the Semantic Web may be theoretically used by Logist as a result.

3. Rigorous evaluations have taken place that demonstrate the benefits of semantic technologies, or validate the results obtained: **A lot of evaluation and validating work has been done but we still cannot guarantee a Rigorous level result, as such, investment, sponsor and corporation are welcomed for a fully functional and quality product to be ultimately presented to end users across the world.**
4. Novelty, in applying semantic technology to a domain or task that have not been considered before: **eying to a legal information system, Logist actually provides support to almost all common information processing tasks, e.g. extensible natural language representation, user machine interaction.**
5. Functionality is different from or goes beyond pure information retrieval: **In addition to translation, contextual editing, statements and entailment rules may be sent to a local data source, etc. are all features that differentiates Logist from a pure information retrieval system.**
6. Contextual information is used for ratings or rankings: **do not apply.**
7. Multimedia documents are used in some way: **Rich style text, e.g. image contained in the description property value in RSS feed are used; Prompt sounds are used interacting with the system; Chart are to be used in data analysis in the future to fully harness the power of JavaFX and ad hoc query.**
8. There is a use of dynamic data (e.g. workflows), perhaps in combination with static information: **hot menu, formalized natural language sentence and sentence pattern as well as RDF data are all dynamic data in combination with static information.**
9. The results should be as accurate as possible (e.g. use a ranking of results according to context): **All data processing is achieved in accuracy and do not need result ranking according to context.**
10. There is support for multiple languages and accessibility on a range of devices: **Logist is a bilingual supportive system which is developed in Java inheritably supportive to multiple platforms as well as multiple languages.**

Instruction on Web access to Logist

Please visit <http://www.pentedge.com/iswc2014/semanticWebChallenge.htm> for detailed information. This page will be available by Sept. 23, 2014 T23:60 Hawaii Time