

OntoQuer: A Tool for Building SPARQL Query Automatically Applying with Our Ontologies

Chochiang K.¹, Grabmann B.¹, Sa-ngiamsak C.², Betbeder M. L.¹, Lapayre J. C.^{1*}

¹ CNRS Femto-ST Institute – UMR 6174 University of Bourgogne France-Comté(UBFC), Besancon, France.

² Department of Electrical Engineering, Faculty of Engineering, Khon Kaen University, Thailand

* Corresponding author. Email: jc.lapayre@femto-st.fr

Manuscript submitted October 1, 2017; accepted December 1, 2017.

doi: 10.17706/jsw.12.2.145-152

Abstract: The aim of this paper is to retrieve results efficiently from 3 related ontologies using a new tool based on SPARQL: “OntoQuer”. OntoQuer consists of 6 menus designed to cover all searches from users. The highlight of OntoQuer is that users can obtain results without any background in SPARQL syntax. OntoQuer compiles at each step in a simple way. The outstanding advantages of OntoQuer are the system can automatically create SPARQL query to find the results and able to validate the correction of syntax. OntoQuer simply proceeds the data in Resource Description Framework (RDF) and the Web Ontology Language (OWL). In addition, the current search with Thai-English transliterated words does not support foreign tourists. The existence of OntoQuer can solve the mentioned problem. The implementation of OntoQuer was designed with 3 newly-developed ontologies (Tourism ontology, Language ontology and UserProfile ontology) to tackle this issue with the case study of Phuket tourism. The ontological knowledge was required in order to improve search effectiveness using ontological data in SPARQL query. The combination of OntoQuer with 3 ontologies has offered a powerful search tool with high efficiency and greater flexibility to users.

Key words: SPARQL, RDF, URI, ontology, OntoQuer.

1. Introduction

In the globalization era, the large amount of information is available on the internet and increasing in number exponentially. Keywords search has several disadvantages; long retrieval time, low accuracy, inability to evaluate the meaning of the user’s query etc., especially, lack of support in transliterated word searching. Tourists may not find suitable and accurate information of touristic attractions on the internet without some help from a personal travel guide or native speakers.

In order to overcome these limitations, semantic web technology is necessary. The Semantic Web is a common framework used to store, presented the structure contents and classified data relationships. It provides the ability to understand meta-data, standard markup language such as RDF or using ontologies and define relations for interoperability [1]. Ontology is a semantic web tool that is being increasingly used for create the applications for the specific domain. Most used ontologies describe individuals (instances), classes (concepts), attributes and relations. Defining ontology in RDF form makes the search system be able to understand the meaning of the information's efficiency; and RDF can gather the knowledge from multiple data sources together.

SPARQL is a popular method for query the relevant records from RDF or OWL file [2]. SPARQL is the W3C candidate recommendation query language for RDF [3]. It is the language that performs fast and efficient search of data. Users must learn SPARQL syntax to query data from ontology. SPARQL contains many capabilities for

querying, including express queries across diverse data sources. Moreover, SPARQL has sensitive syntax and difficult to learn in limitation time. Nevertheless, our new tool enhances the effectiveness of query search using SPARQL. OntoQuer will generate SPARQL automatically from keywords. This is a powerful tool which can reduce the search time and provide more flexibility for users.

In summary, the goal of this work is a new tool called “OntoQuer” which can assist users in building their SPARQL queries. OntoQuer with the new three ontologies : Tourism ontology, Language ontology and UserProfile ontology was tested. The results show that OntoQuer offers a simple and user-friendly interface. OntoQuer is easy to use by only entering keywords to create URIs or literal. This system needs users to select URIs or literal into <subject>, <prediction> and <object>. Then the system will automatically create a complete SPARQL syntax and retrieve accurate and relevant results for their queries. This tool can support and apply SPARQL query with all domain ontologies. Thus, the effective query is very high because the data are linked and interrelated.

The following section presents the related work on RDF, OWL, SPARQL and ontology. Then, in the third section, we introduce our 3 ontologies and OntoQuer. After that, we show the results from querying and conclusion.

2. Related Work

Currently, an increasing amount of linked data and the web of data is available [4]. The representation of the knowledge dedicated to represent the information is in a form that computer system can process to compile the complex systems. There are a variety of language which can be used to represent a conceptual model. RDF (Resource Description Framework), RDFs (RDF Schema) and OWL(Web Ontology Language) are the famous language's representation for the semantic web.

2.1. RDF and OWL

RDF is a standard data model to represent information about resources in the World Wide Web (i.e. meta data). RDF identifies the relationship between nodes that link the structure of the Web using URI [5].

The Uniform Resource Identifier (URI) is a string of characters that identifies a unique identification for a resource. It allows interacting with the representations of the resource on the internet, typically WWW (World Wide Web) [6]. Ontology is a semantic web tool for interactions in some particular domain of knowledge. Ontologies have several advantages such as interoperability, sharing, reuse...[7]. OWL is a semantic web language that provides syntax and represents the knowledge and the relations between things for the semantic web. It also provides the solution for limitations in terms of identifying the relationships between classes in RDF.

The fundamental build up on each of these languages is the triple. It contains three components:subject, predicate and object¹. Subject, Predicate and Object must be identified by URIs, or by another Object who can be a constant data value (or literal) instead of a URI. Subject is the resource that described by the statement. Predicate is also known as the property of the triple. Object is the specific value of the property. OWL can be used to mix between RDF and XML (Extensible Markup Language) depending on the methodology and the information that are created in .owl, .rdf and .xml [8].

2.2. SPARQL (Search Protocol and RDF Query Language)

Using query language can reduce the time obviously for searching and retrieving information. The semantic web ontologies code is based on RDF query language (including RDFs and OWL). There are many existing semantic web query language such as RQL, RDQL, Versa and SPARQL [9]. SPARQL improved searching the data faster and efficient when compared to other Query language, was presented by [2]. In addition, SPARQL is the query language recommended by W3C (World Wide Web Consortium) currently [3]. SPARQL query is executed and supported by the most RDF triple store [10]. SPARQL works are based on comparison between RDF and query statement. If the statements are matched, the RDF results will be shown.

The principle process has two parts that are SELECT and WHERE. The first part identifies the triple values that will be displayed from searching on the RDF graph. The second part recognizes the results of which conditions

matched the RDF datasets. SPARQL has many benefits over the other query languages. For example, SPARQL is an appropriate query language for querying two or more data sources in a single query. SPARQL can query RDF datasets natively. SPARQL can support for querying semi-structured, unpredictable and unreliable structure. It is able to use the various queries to retrieve the output and it can query the RDF graph that consists of various triples. And it helps to reduce time in query language and retrieve fast.

¹ Three Triples : [Available <http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/#section-triples>]

There are several works have proposed for SPARQL queries. Nandish Jayaram et al., [11] created the graphical systems to query the knowledge using GQBE (Graph Query by Example) It can find the best answers with fast query processing. Geraud Fokou et al., [12] constructed graphical tool for SPARQL query too, but this tool can improve to support query-answering technique. In order to make SPARQL queries more easily, [13] proposed a web-based prototype based on the selected SPARQL for users who never used (Visual-OWL) before.

2.3. Ontology

Ontology constitutes the standard knowledge representation tool for the Semantic Web. It is an efficiency tool for interactions in some particular domain of knowledge. Ontology provides an explicit conceptualization that describes the semantics of the data. Ontology can support the sharing common understanding of the domain and related information among users or software agents. Reusing ontology makes getting new ontologies which are more quality because the reused components have been tested already.

3. Contribution

3.1. Our Platform: WICHAI

WICHAI (WebservICes platform for pHuket tourism bAsed on ontologies) is a platform for searching touristic information (attractions, activities, restaurants, . . .) in Phuket based on semantic web technology [14]. There are web-based application, ontology and transliteration (shown in Fig. 1). The main point for this paper is ontology part.

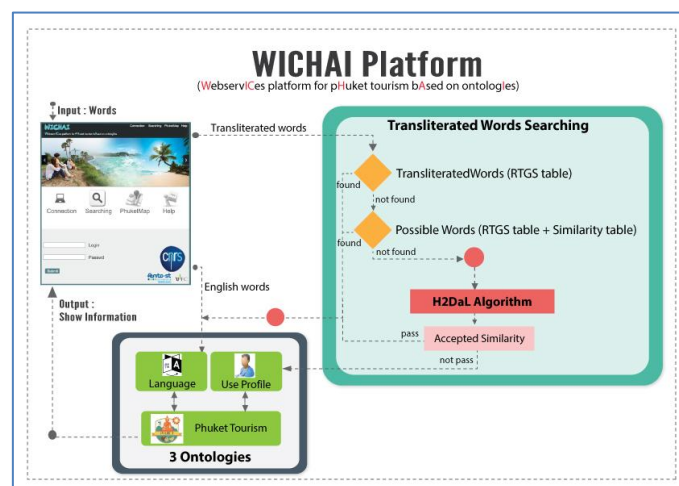


Fig. 1. WICHAI platform.

Ontology development is rapidly evolving in all domains. The obvious advantage of ontologies is supporting the sharing and reusing the knowledge. Applying the reused ontologies helps to save the budget, resource and time. The important reason to apply reused ontology because they had already been tested. In this work, ontology consists of 3 domains that are Tourism, UserProfile, and Language ontologies. Tourism ontology and UserProfile ontology are created by reusing the existing ontologies and extending some new super-class and sub-class. The last ontology, language ontology, we create the new one because the existing language ontologies are not suitable for Transliterated words in Thailand. These ontologies have already been validated by 3 experts

in the field of tourism in Phuket and ontology management from Thailand. Our ontologies are shown in Fig. 2.

Tourism plays an important role in searching information. Currently, The Tourism Authority of Thailand promotes the “Chom-Chim-Chop” campaign. “Chom” interprets visiting, “Chim” means eating and “Chop” is shopping activity. Thus we create and reuse domain of ontologies based on the government slogans. Tourism ontology is adapted from ETP ontology [15] and OnTour ontology [16]. ETP is a management system of destination for tourism services online such as identifying, classifying, organizing, ... by the United Nations Conference on Trade and Development e-tourism initiative. OnTour is a Tourism ontology domain using OWL. It is based on the Thesaurus on Tourism and Leisure Activities of the World Tourism Organization. However, existing ontologies do not cover exactly the concepts related to tourism in Phuket, Thailand. This became the motivation for this work: the new one based on ETP ontology and OnTour ontology. Tourism ontology contains eight classes based on “Chom-Chim-Chop” slogans. All classes are created in .owl file.

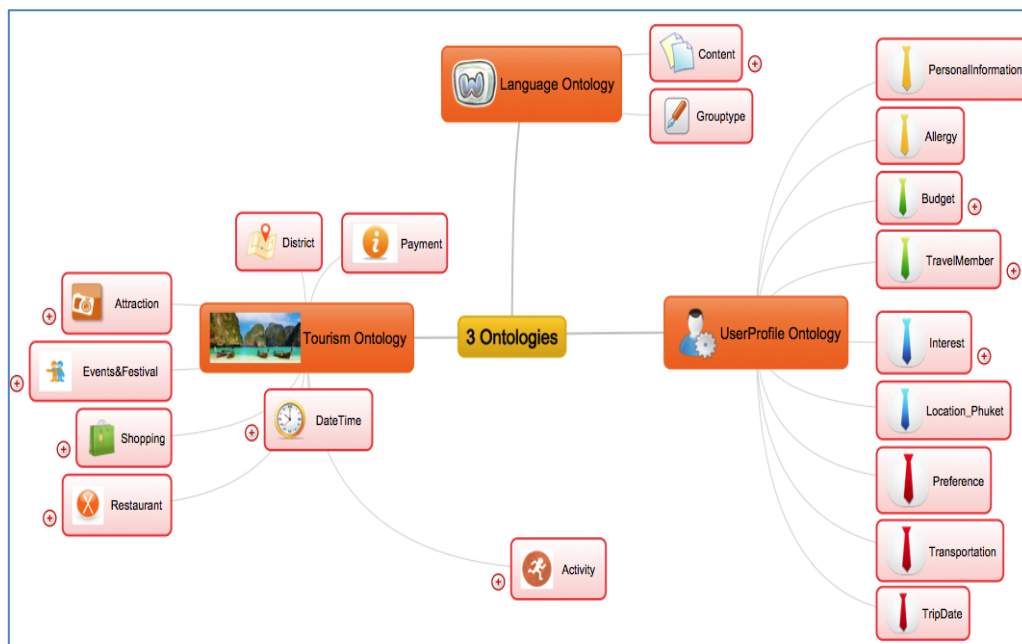


Fig. 2. Combination of ontologies.

The UserProfile ontology aims to facilitate the extraction of the details of the user personal information. This UserProfile ontology is based on Tamilnadu ontology [17] which are Indian scenario (Tamilnadu) tourism. Tamilnadu was created to aid the tourists to plan their trips accurately. User Profile ontology is very important. It is the first step to classify data properties based on user preferences. This ontology collects various details such as preferences, budget, date of the trip, ... There are total 9 classes and there are object properties, data properties, instances, ...

The language ontology manages and groups words into English words, Transliterated words and possible words. In addition, this ontology has class for collecting synonym words to provide more powerful searching. Our 3 ontologies are developed using protégé. System based on collaborating 3 ontologies can provide tourism's information that match the user's query based on linguistics and user profile effectively.

From 3 related ontologies, SPARQL query is the important method to retrieve the correct results. But SPARQL has limitations of usage. Thus, we proposed “OntoQuer” to solve this issue.

3.2. Our Tool: Onto Quer

In order to make SPARQL queries more accessible to users, OntoQuer tool was developed by Java programming language, Swing and AWT libraries using Eclipse. It is suitable for beginner who wants to design, create the

SPARQL query without the knowledge of complex syntax and the structure of the ontology. It is the tool to generate URIs and users can select URIs into <subject><prediction><object>. Then, the SPARQL query will be automatically created. This system has various and comprehensive menus such as simple queries lists, lists of entities, classes, ... stored by the user. The user interface consists of 6 main menus are show in Fig. 3.

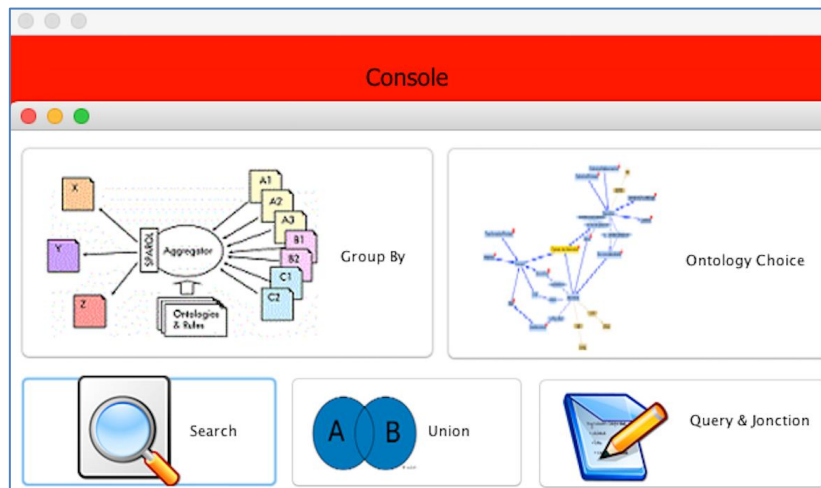


Fig. 3. User ontoquer's user interface.

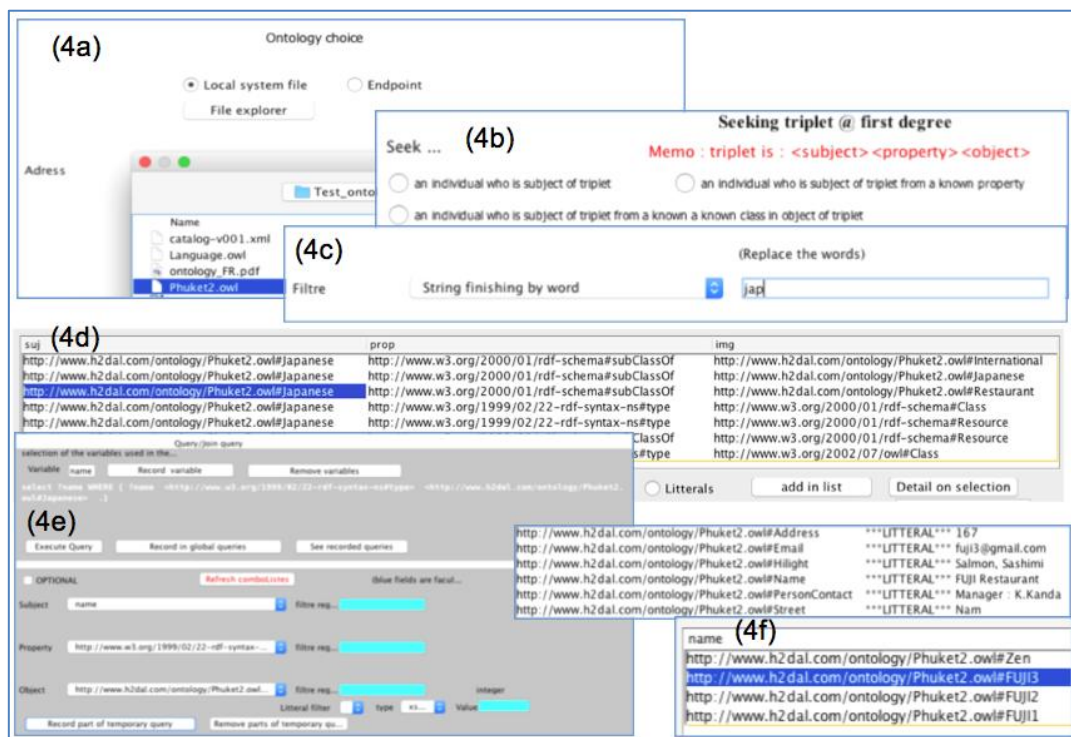


Fig. 4. User interface in each step of ontoquer.

The first menu is "Ontology Choice", users can select the ontologies and import the single ontology or multiple ontologies into the system. Next is "Search" function. The users insert the keywords for classes, properties, individual from subject or object part, ... for simple filtering, and the system will generate URIs automatically. Third is "Query&Inner Join", users can select list of <subject>, <property> and <object>. This process will build the SPARQL query using URIs from search function. Fourth process is union function and next is grouping functions such as sum, count, ... Last is the console window in order to display some messages from the system

such as redirect errors, message for connection, ...

OntoQuer is a tool for users creating and reviewing SPARQL query easily. It has various advantages as following: It is user-friendly, even with only one or multiple ontologies. Users do not require the knowledge of SPARQL syntax. This tool allows the users to validate completely or redundancy of ontologies structure. The results are clear visualization, easy to be understood. It can show the details as needed; and it is able to determine the accuracy of the ontology structure. Easy searching process, users can search in each step respectively.

For example, users would like to “search location and opening time for Japanese Restaurant from our ontologies”. Beginning with choosing and importing the ontologies, then selecting the part of triple from class, property, individual, ... for create URIs (Fig. 4a and Fig. 4b). Then, users insert the keyword from the question, users can select keywords (Fig. 4c). After that, the system will create URIs and users are able to save each URIs into class, property, individual, ... for next queries (Fig. 4d). Moreover users can select the interested URIs for more information. Then, Query and Inner Join part, users select URIs or literal into <subject><property><object> from combo-list. The system will create and show the SPARQL query automatically (Fig. 4e). After user select “execute query” button, the answers will be displayed from SPARQL query (Fig. 4f).

4. Results

OntoQuer was applied with our 3 ontologies. The results clearly show that the structure of those 3 ontologies are correct using SPARQL query generated by OntoQuer. From question in section 3.2, the result of Japanese Restaurant was displayed as shown in Fig. 4. Japanese Restaurant is the subclass of International Restaurant and Restaurant respectively. Users can enter the keyword only “jap”, “japa”, “japanese”, The tool will create the URIs to users automatically. Querying part, users simply select the value or URIs that was recorded previously according to <subject><prediction><object> appropriately. Finally, the system will generate the SPARQL query and offer all Japanese Restaurants from the ontologies. Moreover, users can also get in-depth details such as location, opening hour, ... And users can query with many conditions such as “Where is the places that users can buy the souvenirs on Friday at 9.00 PM?” The system creates the UNION relationship suitable for the user's requirement.

In addition, supporting for query with various ontologies is another important feature. It connects the data in each domain by using simple SPARQL queries. From Fig. 2, Please retrieve the answer for this question “Who (data from UserProfile ontology) love monkey (data from Tourism ontology)? And please suggest the place whether she should go in English words and Transliterated words (data from Language ontology)”. This question consists of 3 phases, the first step to find the keywords “monkey”. Then, the output shows “Nana” who is the person who love monkey from UserProfile ontology (see Fig. 5(above)). After that, the system will show description about “Zoo” that it is the place for animals from Tourism Ontology. The last step is connecting with Language ontology to show that zoo is “Suansat” or “Sat” in transliterated word. (See Fig. 5(bottom))

http://www.h2dal.com/ontology/UserProfile.owl#Depart	***LITERAL*** 2016-04-10T06:00
http://www.h2dal.com/ontology/UserProfile.owl#Dislike	***LITERAL*** adventure activity
http://www.h2dal.com/ontology/UserProfile.owl#Like	***LITERAL*** Outdoor activity, monkey
http://www.h2dal.com/ontology/UserProfile.owl#Name	***LITERAL*** Nana
http://www.h2dal.com/ontology/UserProfile.owl#NumberOfAdult	***LITERAL*** 2
http://www.h2dal.com/ontology/UserProfile.owl#NumberOfChildren	***LITERAL*** 1
prop	img
http://www.h2dal.com/ontology/Phuket2.owl#hasGroup	http://www.h2dal.com/ontology/Phuket2.owl#Attraction
http://www.h2dal.com/ontology/Phuket2.owl#words_EN	***LITERAL*** zoo
http://www.h2dal.com/ontology/Phuket2.owl#words_PO	***LITERAL*** Suansud, Suansut, Sud, Sut
http://www.h2dal.com/ontology/Phuket2.owl#words_TR	***LITERAL*** Suansat, sat

Fig. 5. User Interface for the results.

Above query results, the users do not require the knowledge of query syntax. Users can search by only putting keywords and understand each step in the searching process. This tool can process the variety of different functions such as grouping and aggregation, union, ... The last advantage of this tool is its suitability for beginners with less knowledge in SPARQL syntax using OntoQuer to confirm the correct syntax.

5. Conclusion

Currently, ontology is an important conceptualization covering a variety of information sources. The datasets can group and correlate with the different domain ontology. There are several methods and implementations for query data under RDFs. Using SPARQL to query ontologies and metadata are a popular method. Unfortunately, SPARQL syntax is difficult to learn because it is sensitive and strict. If users type wrongly (even only one alphabet), the results cannot be displayed. Thus, most users avoid using this method. Finally, we propose a new tool for building SPARQL query. It is easily used by users who have low knowledge in syntax of SPARQL. The user interface is simple and easy. It defines the category and supports from various ontologies. When users search the queries from 3 ontologies using OntoQuer by providing only keywords, the results will show the best recommended attractions in Phuket based on user profile.

For the future work, we plan to conduct a recommender system. It integrates a search engine, 3 ontologies being validated by OntoQuer and our algorithm part. Using SPARQL connected with the ontologies via Jena (Java API) will be explored. Then we combine SPARQL query from OntoQuer with the recommender system to get the best results. We will also develop OntoQuer into web-based application in order to have more efficient use. It will be more easily to use OntoQuer directly to searching that gives more accurate and efficient results.

References

- [1] Issara, C., *et al.* (2014). Information retrieval in thai northeast travel utilizing ontology. *Information Technology Journal*, 10, 15–25, 2014.
- [2] Sheba, P. A., *et al.* (2012). An enhanced method for efficient information retrieval from resume documents using SPARQL. *Data Mining and Knowledge Engineering*, 4(1), 5–10.
- [3] SPARQL query language for RDF. Retrieved from: <http://www.w3.org/TR/rdf-sparql-query/>
- [4] Aynaz, T., & Mehrnoush, S. (2012). Consolidation of linked data resources upon heterogeneous schemas. *Proceedings of the Sixth International Conference on Advances in Semantic Processing*.
- [5] Jorge, P., *et al.* (2009). Semantics and complexity of SPARQL. *ACM Transactions on Database Systems (TODS)*, 34(3).
- [6] Tim, B. L., *et al.* (2016). Uniform resource identifier (uri): Generic syntax. Retrieved from: <https://tools.ietf.org/pdf/rfc3986.pdf>
- [7] Zuoshuang, X., *et al.* (2010). Ontofox: web-based support for ontology reuse. *BMC Research Notes*, 3(1), 175.
- [8] Naruepon, P., & Chakkrit, S. (2011). System development of search tourism ontology with sparql. *Proceedings of the 7th Naresuan Research Conference*.
- [9] Landon, D. *et al.* (2008). Regular paths in sparql: querying the nci thesaurus. *Proceedings of the AMIA: Symposium Proceedings*.
- [10] Ilianna, K., *et al.* (2011). SPARQL query answering over owl ontologies. *The Semantic Web: Research and Applications*.
- [11] Nandish, J. *et al.* (2015). Querying knowledge graphs by example entity tuples. *IEEE Transactions on Knowledge and Data Engineering*, 27(10).
- [12] Géraud, F., *et al.* (2015). Qars: A user-friendly graphical tool for semantic query design and relaxation.
- [13] Florian, H., *et al.* (2015). Queryvowl: Visual composition of SPARQL queries.
- [14] Kitsiri, C., *et al.* (2016). A new web-services based platform for semantic transliterated words searching to facilitate tourists trips in Thailand. *Proceedings of the 2016 International Conference on Computer Supported Cooperative Work in Design*.
- [15] Hafedh, M., *et al.* (2011). E-tourism portal: A case study in ontology-driven development. In *E-Technologies: Transformation in a Connected World*.
- [16] Kathrin, P. (2005). Ontour ontology. Retrieved from: [e www.deri.org](http://www.deri.org)
- [17] Ananthapadmanaban, K. R., & Srivatsa, S. K. (2011). Personalization of user profile: Creating user

profile ontology for tamilnadu tourism. *International Journal of Computer Applications*.



Kitsiri Chochiang is a PhD student at the CNRS FEMTO-ST Institute (Computer Science Dept) in Bourgogne/Franche-Comte University (France). She is working especially in the Department of Computer Science for Complex Systems.

Her current research focuses on ontology, linguistic and web service. From this work she developed an interest ontology-based reasoning and its applications in Tourism using Thai-English transliteration to search effectively.



Bertrand Grabmann is a student in research master's degree at the CNRS FEMTO-ST Institute (Computer Science Dept) in Bourgogne/Franche-Comte University (France) and associate of SOFIMEX a firm based near Lille (France) and specialised in commercial statistics for customs.

Since 18 years, his main domain is technical architecture and development of information systems. He is also interested in facilitating SPARQL queries in ontologic systems.



Chiranut Sa-ngiamsak received Ph.D in electrical engineering and electronics from University of Manchester Institute of Science and Technology (UMIST), England in 2004. Since 1994, she has become a full-time lecturer at Department of Electrical Engineering, Faculty of Engineering, Khon Kaen University, Thailand where she is now holding an associate professorship.

Her research interests are high speedsolid state device and memory technology. She is now actively working on failure analysis of memory technology and solid-state semiconductor design.



Marie-Laure Betbeder is an associate professor at the CNRS FEMTO-ST Institute (Computer Science Dept) in Bourgogne/Franche-Comte University (France). She is working especially in the Department of Computer Science for Complex Systems.

For the last ten years, she has been involved in the domains of technology enhanced learning and computer supported collaborative learning. Her early works were focused on the taylorability of computer artifacts as a support for learning environments. She is now working on the use of ontologies for the CSCW domain (computer supported collaborative

work).



Jean-Christophe Lapayre is professor at the CNRS FEMTO-ST Institute (Computer Science Dept) in Bourgogne/Franche-Comte University (France) since 2002. He is responsible of distributed algorithmic for Teleapplications Research Group, and member of the Biom'@x Steering Committee (FEMTO-ST's research axis on biomedical area).

His general field is the distributed systems and his present research interests are on ontologies used in distributed collaborative platforms. He is the Head of the third year in ISIFC Biomedical Superior Institute of Engineers of Franche-Comte with a specific e-Health & Telemedicine orientation.