Expert Systems with Applications 40 (2013) 7428–7443

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Semantic Ontology Mapping for Interoperability of Learning Resource Systems using a rule-based reasoning approach

Ngamnij Arch-int *, Somjit Arch-int

Semantic Mining Information Integration Laboratory (SMIIL), Computer Science Department, Science Faculty, Khon Kaen University, Khon Kaen 40002, Thailand

ARTICLE INFO

Keywords:
Ontology mapping
Learning resource systems
Semantic interoperability
Learning object
Semantic conflicts
Structural conflicts
Rule-based reasoning

ABSTRACT

There is an increasing demand for sharing learning resources between existing learning resource systems to support reusability, exchangeability, and adaptability. The learning resources need to be annotated with ontologies into learning objects that use different metadata standards. These ontologies have introduced the problems of semantic and structural heterogeneity. This research proposes a Semantic Ontology Mapping for Interoperability of Learning Resource Systems. To enable semantic ontology mapping, this research proposes conflict detection and resolution techniques for both semantic and structural conflicts. The Semantic Bridge Ontology has been proposed as a core component for generating mapping rules to reconcile terms defined in local ontologies into terms defined in the target common ontology. This work defines the reasoning rules to classify related learning objects to enhance the powerful deductive reasoning capabilities of the system. As a consequence, ontology-based learning object metadata are generated and used by the semantic query engine to facilitate user queries of learning objects across heterogeneous learning resource systems.

1. Introduction

Many Web-based educational systems have been developed to meet the growing needs of the education community for e-learning efficiency, flexibility, and adaptability. These systems usually offer an abundance of learning resources presented in any form of digital learning material, such as document files, web pages, audio, video, and even image files. Finding the relevant learning resources requires that these resources be annotated with rich, standardized, and widely used metadata. Although several educational metadata standards, such as IEEE LOM (IEEE, 2002), Dublin Core (DCMI, 1990), IMS consortium (IMS, 2001), and SCORM (ADL, 2004), have emerged to define a standard specification of a learning resource component as a learning object to facilitate accessibility, interoperability, and reusability, the use of different types of metadata has introduced the problem of interoperability across heterogeneous learning resource systems.

Our main motivation lies in tackling two main kinds of problems:

(1) *The lack of a semantic common standard in the e-learning domain:*
The abundance of e-learning metadata standards and user-defined metadata does not propose a globally agreed common standard and still lacks formal semantics. For the new Web generation, the Semantic Web provides an ontology for organizations to annotate learning resources semantically into learning objects to facilitate knowledge sharing and reuse by others. The ontology enables the organization to annotate learning resources semantically into learning objects to facilitate knowledge sharing and reuse by others. As the number of learning objects increases, it can be expected that many different ontologies will appear to annotate each learning resource. These different ontologies can use different e-learning metadata to describe the same or similar sets of learning objects. The related learning objects dispersed over the heterogeneous learning resource systems are still difficult to discover and reuse within other courseware, because of the lack of a semantic common standard between learning resource ontologies. To access learning objects, users need to know the physical locations of the desired learning objects. This leads to the problems of interoperability of heterogeneous learning resource systems.

(2) *The occurrence of conflict types of ontology mapping:*
The occurrence of different learning resource ontologies must be reconciled through ontology mediation to enable semantic interoperability. A number of studies (such as Alcaraz Calero et al., 2010; Jean-Mary, Shironoshita, & Kabuka, 2009; Martínez-Gil, Delgado-Navas, & Aldana-Montes, 2012; Rodriguez & Egenhofer, 2003) have relied heavily on

* Corresponding author. Tel.: +66 43 342477; fax: +66 43 342910.
E-mail addresses: ngamnij@kku.ac.th (N. Arch-int), somjit@kku.ac.th (S. Arch-int).

0957-4174/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2013.07.027

resolving semantic conflicts, leaving aside the structural conflicts that usually exist when multiple ontologies must be interoperated. Since the structural conflicts can be classified into several levels of conflict types, the structural conflicts resolution is rather complex and is a daunting task. There is still need for automatic or at least semi-automatic techniques to cope with these kinds of conflict to reduce the burden of manual creation and maintenance of mappings. At the same time, the conflicts resolution approach should be flexible enough to keep independent of domains and extend to other kinds of conflict. Because the Semantic Web provides a very powerful rule-based reasoning system to enhance deductive reasoning capability, this research envisions that the rule-based reasoning approach is an interesting alternative to defining such mappings of multiple ontologies.

1.1. Contribution of this paper

This research proposes the **S**emantic Ontology **M**apping for **I**nteroperability of **Le**arning Resource Systems, called the SMILE, to support semantic ontology mapping and semantic querying of heterogeneous learning objects. The SMILE project aims to resolve two main kinds of above problems by addressing the following issues:

(1) To cope with problem one, this work has adopted the common learning object ontology (the so-called CO) expressed in Web Ontology Language (OWL) as a common ontology incorporating most common metadata schemes such as IEEE LOM and the Dublin core. The CO provides an abstract view of learning resource vocabularies on which users can pose their requests over the terms defined in the CO. Thus, the underlying heterogeneous learning resources are concealed from the users. In addition, this work also defines reasoning rules (Jena, 2010) to classify related learning objects to enhance the powerful deductive reasoning capabilities of the query processing system and to enable interoperability of heterogeneous learning resource systems.
(2) To cope with problem two, this research lies in the conflict detection and semantic mapping techniques. To enable conflict detection, this work proposes conflict detection rules which enable a systematic detection approach to both semantic and structural conflicts. To enable conflict resolution, this work proposes the Semantic Bridge Ontology Mapping tool to generate the Semantic Bridge Ontology (or SBO) by providing a mapping interface to map the terminologies of different local ontologies to a common set of ontology terminology defined in CO. The SBO is then used as a core component to generate mapping rules to transform local ontology instances into instances in the target CO. Thus, the CO is also used to reconcile the semantic and structural discrepancies occurred in different learning resource ontologies. The mapping rules use a rule-based reasoning language as the core mechanism for conflict resolution.
(3) The proposed approach provides an application for managing learning objects and supports semantic querying of learning objects through the CO terminology. This ensures that the proposed approach can be applied to a practical case to validate the viability of model realization.

1.2. Structure of the paper

The remainder of the paper is structured as follows: Section 2 presents a literature review. Section 3 presents the SMILE architecture. Section 4 proposes the CO structure and definition. Section 5 describes motivating examples and heterogeneity problems. Section 6 presents the semantic ontology mapping process. Conflict detection and resolution are presented through algorithms and mapping rules generation. The reasoning rules to classify related learning objects are also defined. Section 7 presents the SMILE system implementation, Section 8 presents contributions and comparison with other approaches, and Section 9 summarizes the work.

2. Literature review

2.1. Exploiting the Semantic Web in the e-learning domain

The goal of Semantic Web (Breitman, Casanova, & Truszkowski, 2007) is to declare explicitly the knowledge embedded in many Web-based systems, to enable computers to understand Web content, and to integrate information in an intelligent way. The Semantic Web relies heavily on formal ontologies for achieving knowledge management by capturing, sharing, and reusing knowledge within a specific domain (Beydoun, Low, Tran, & Bogg, 2011; Ferrario, Guarino, & Prévot, 2007; Santoso, Haw, & Abdul-Mehdi, 2011), which is necessary to overcome differences in terminology in a domain. Creating a machine-understandable ontology for a computer or software agent requires that the ontology structure be transformed into a language form such as the Resource Description Framework (RDF) (Brickley & Guha, 2004) or the Web Ontology Language (OWL) (Smith, Welty, & McGuinness, 2003). To enhance the powerful deductive reasoning capabilities of the system, the Semantic Web along with rule languages will enable a machine to synthesize new facts which will be stored in the knowledge base (Biletskiy, Boley, Ranganathan, & Boley, 2008; Chen, 2010; Saake, Sattler, & Conrad, 2005). Examples of such rule languages that are widely used in Semantic Web applications are Datalog (Ullman, 1988), the Rule Markup Language (RuleML), the Semantic Web Rule Language (SWRL) (Horrocks et al., 2004), and Jena rules (Jena, 2010).

Therefore, the Semantic Web is a prominent technology and very suitable for use in the e-learning domain in various ways. Several researchers (such as Brut, Sedes, & Dumitrescu, 2011; Chu, Chen, & Chen, 2009; Cuéllar, Delgado, & Pegalajar, 2011b; Fernández-Breis et al., 2012; Yaghmaie & Bahreininejad, 2011) have used ontologies to annotate learning resources semantically and to perform semantic querying for suitable learning material. These studies (such as Cuéllar, Delgado, & Pegalajar, 2011a; Muñoz-Merino, Kloos, & Naranjo, 2009) have focused on semantic interoperability of different e-learning systems and on providing a common semantic framework for data sharing between different LMS databases. In the context of learning objects interoperability, Lee, Tsai, and Hsieh proposed a knowledge exchange framework and a set of semantic inference rules that can leverage interoperability among semantically heterogeneous learning objects (Lee, Tsai, & Hsieh, 2011). Chi provided an ontology to represent abstract views of content sequencing and course materials and added semantic rules to represent relationships between individuals (Chi, 2009). Knight, Gašević, and Richards used ontologies to specify explicitly all learning designs, all learning objects, and the relations between them, and proposed more effective (semi-)automatic tools and services to increase the degree of learning object reusability (Knight, Gašević, & Richards, 2006). In other studies, (such as I-Ching, 2012), the Semantic Web technologies have been integrated into a common framework called the Multi-Layered Semantic LOM Framework to facilitate machine understanding and to enable intelligent discovery of learning objects.

2.2. Ontology mediation

In the context of semantic knowledge management, many different content providers use different ontologies to annotate their content because it is hard to agree on a common vocabulary for a large, spatially distributed group. To enable inter-operation between different ontologies, ontology mediation is especially important to enable sharing and reuse of data. Several researchers (such as Choi, Song, & Han, 2006; Bruijn et al., 2006; Kalfoglou & Schorlemmer, 2003; Kaza & Chen, 2008) have surveyed ontology mapping and merging theories, including techniques, methods, and tools. This paper includes a survey of ontology mapping techniques applied to several research areas. Several studies (such as De Souza, Moura, & Cavalcanti, 2010; Jung, 2010a, 2010b; Seng & Kong, 2009) have used ontology mapping techniques to enable system interoperability. In the context of resolving semantic conflicts in information integration, Jean-Mary, Shironoshita, and Kabuka proposed automated semantic matching of ontologies using a verification algorithm that uses the lexical and structural characteristics of two ontologies to calculate a similarity measure between them iteratively, to derive an alignment, and then to verify whether it contains semantic inconsistencies (Jean-Mary et al., 2009). Some research studies (such as Schorlemmer & Kalfoglou, 2008) have provided a mathematical ontology-based semantic integration framework to cope with both the terminology and the interpretation mechanism of the semantic heterogeneity problem. Chungoora and Young proposed a common logic-based approach (Chungoora & Young, 2011) to achieve semantic reconciliation in the context of the Semantic Manufacturing Interoperability Framework (SMIF). The approach aimed to enable formalization and verification of cross-model semantic correspondences in design and manufacture by applying the relevant logic-based mechanisms. In the context of semantic learning resources discovery, Yu, Zhou, and Shu introduced a context-aware e-learning infrastructure called the Semantic Learning Space (Yu, Zhou, & Shu, 2010) based on Semantic Web technology. They used an ontology-mapping technique to achieve semantic content integration by transforming different kinds of content-description metadata into generic ontology mark-ups to enable semantic queries across heterogeneous contents. The types of learning content that are suitable for various user contexts are recommended using fuzzy theory. Gašević and Hatala proposed an ontology mapping-based framework that enables searching for learning resources using multiple ontologies (Gašević & Hatala, 2006). The framework used SKOS (Simple Knowledge Organization System) (Miles & Bechhofer, 2009) to build an ontology of the ACM CCS and of specific course content. The SKOS mapping properties were also used to create mapping relations between concepts from different domain ontologies.

The comparison of the SMILE approach with the other approaches will be described in Section 8.

3. Architecture of Semantic Ontology Mapping for Learning Object Interoperability

This section presents the **S**emantic Ontology **M**apping for **I**nteroperability of **Le**arning Resource Systems (SMILE) architecture, as shown in Fig. 1, to support the semantic ontology mapping and semantic querying of learning objects through common ontology terminologies. SMILE provides a Semantic Bridge Ontology (SBO) mapping-tool interface for importing the learning resource ontologies (LROs) and the common ontology (CO). The LROs provided by learning resource providers are extracted and annotated independently from different learning resource repositories. These LROs may not agree on a common vocabulary and may involve different semantic terminologies and structural conflicts. The CO developed by the ontology engineer provides an abstract view of learning resource vocabularies on which users can pose their requests over the terms defined in the CO and hide the underlying heterogeneous learning resources from the users. To cope with semantic and structural conflicts, the SBO mapping tool provides an ontology-mapping interface for the ontology engineer to bridge the LRO terminologies to terms defined in the CO. As a result, the SBO is initialized and can be imported to the rule-based ontology mapping (ROM) engine, which generates the mapping rules to transform LRO instances into CO instances. The ROM engine also provides an interface for the ontology engineer to create additional reasoning rules to enhance the powerful deductive reasoning capabilities. As a consequence, ontology-based learning object metadata (OLOM) are generated through the semantic ontology mapping and reasoning process and are used by the semantic query (SQ) engine of the SMILE system to make it easier for users to query learning objects across different learning resource repositories that can leverage interoperability among heterogeneous learning resource systems. Details of CO, LRO, and SBO structures are described in the following sections.

4. The common learning object ontology structure and definitions

This section proposes an e-learning common learning object ontology structure (called the CO) as a common ontology on which users can pose their requests over the terms used in the CO. Other local ontologies with different learning resource vocabularies must be translated and extended to the CO terminology. For the first process of CO construction, the initial CO needs to be constructed. However, the CO structure can be incremented through an evolutionary process while conflicts are detected and need to be resolved. The generic CO structure can be introduced on the basis of ontology components which are defined as a general notation based on object-oriented and set theory as follows:

Definition 1. A common learning object ontology, denoted as CO, is defined as a tuple:

$$\boldsymbol{CO} = \langle \boldsymbol{C_{co}}, \boldsymbol{P_{co}}, \boldsymbol{I_{co}}, \boldsymbol{RC_{co}}, \boldsymbol{RP_{co}}, \boldsymbol{\sigma_{co}} \rangle,$$

where $C_{\boldsymbol{co}}$, $P_{\boldsymbol{co}}$, $I_{\boldsymbol{co}}$, $RC_{\boldsymbol{co}}$, $RP_{\boldsymbol{co}}$, and $\sigma_{\boldsymbol{co}}$ are defined as follows:

Definition 2. $C_{\boldsymbol{co}}$ is a finite set of distinct common concepts (or classes), $C_{\boldsymbol{co}} = \{c_{ci} | \forall i = 1 \ldots n\}$.

Definition 3. $P_{\boldsymbol{co}}$ is a finite set of distinct common properties, $P_{\boldsymbol{co}} = \{p_{ci} | \forall i = 1 \ldots n\}$.

Definition 4. $I_{\boldsymbol{co}}$ is a finite set of all distinct individuals (or instances) which are members of common concepts, $I_{\boldsymbol{co}} = \{_{fci} \in c_{cj} | \forall i = 1 \ldots n, \forall j = 1 \ldots m\}$.

Definition 5. $RC_{\boldsymbol{co}}$ is a finite set of all relationships between common concepts and is defined as a set of triplets, $RC_{\boldsymbol{co}} = \{c_{cm}, r_c, c_{ck})| c_{cm}, c_{ck} \in C_{\boldsymbol{co}}$ *and* $r_c \in \acute{R}_c\}$, where $\acute{R}_c = \{$*subClassOf*, *disjoint*, *union*, *equivalentClass*$\}$.

Definition 6. $RP_{\boldsymbol{co}}$ is a finite set of all relationships between common properties and is defined as a set of triplets, $RP_{\boldsymbol{co}} = \{(p_{cm}, r_p, p_{ck}) | p_{cm}, p_{ck} \in P_{\boldsymbol{co}}$ and $r_p \in \acute{R}_p\}$, where $\acute{R}_p = \{$*subpropertyOf*, *equivalent Property*$\}$.

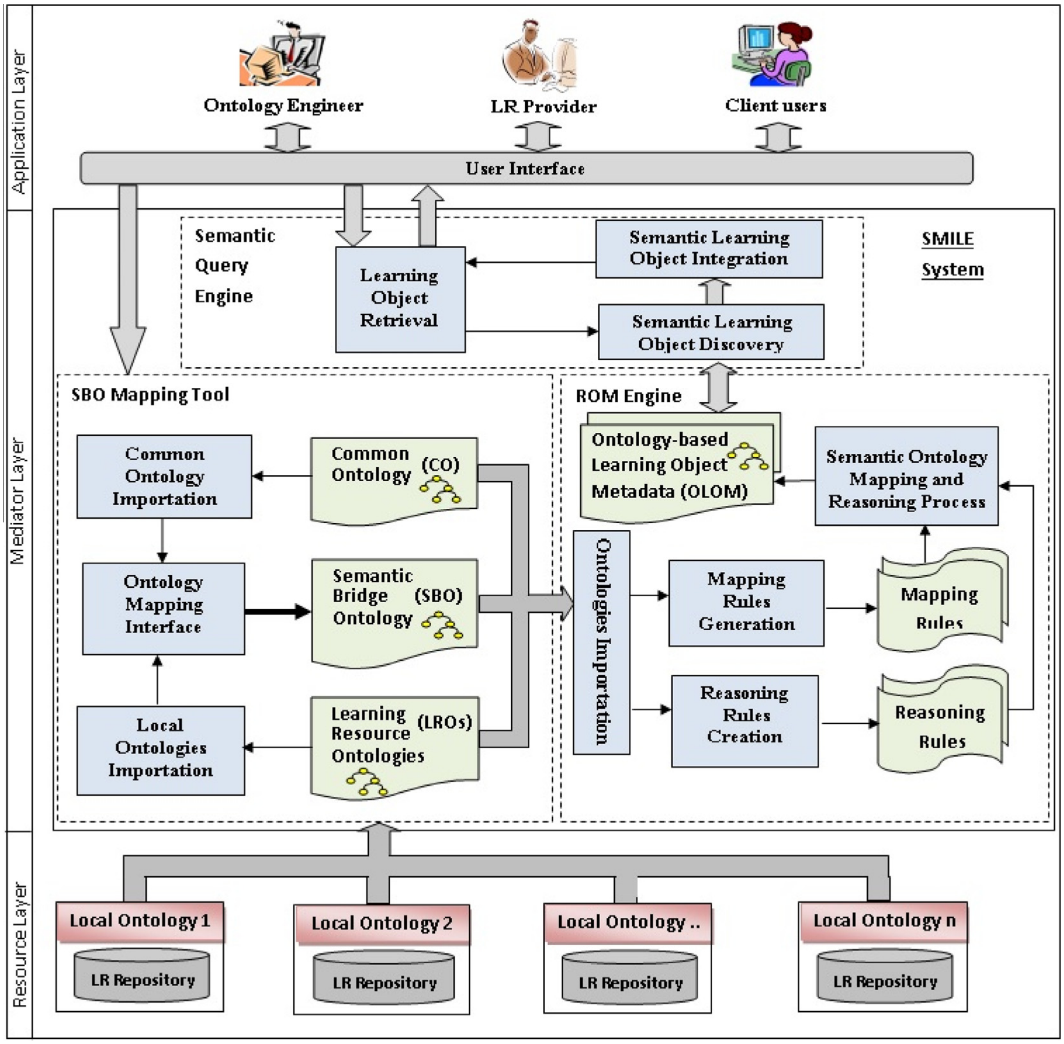


Fig. 1. Semantic Ontology Mapping for Interoperability of Learning Resource Systems Architecture.

Definition 7. σ_{co} is a finite set of property relationships that relate subjects and objects of the properties and is defined as a set of triplets, $\sigma_{co} = \{(s_{ck}, p_{ck}, o_{ck})|p_{ck} \in P_{co}, s_{ck} \in C_{co}$, or I_{co} and $o_{ck} \in C_{co}$ or I_{co}, or $o_{ck} = Literal\ value\}$, where p_{ck} is called a property relationship, s_{ck} is called a subject of property p_{ck}, and o_{ck} is called value of property p_{ck}.

To define domains, ranges, and types of common properties, the following functions are available:

$\bar{D}$: $P_{co} \rightarrow C_{co}$ gives the set of domain concepts (C_{co}) of a property $p_{ck} \in P_{co}$;
$\check{R}$: $P_{co} \rightarrow O_{co}$ gives the set of range concepts or literal (O_{co}) of a property $p_{ck} \in P_{co}$;
Γ: $P_{co} \rightarrow T_p$ gives the set of property types (T_p) of a property $p_{ck} \in P_{co}$;
If ($\check{R}(p_{ck}) = o_{ck} \in C_{co}$), then $\Gamma(p_{ck}) = ObjectProperty$. If ($\check{R}(p_{ck})$ = Literal), then $\Gamma(p_{ck}) = DatatypeProperty$.

Definition 8. Let $ST = \{v_i|\forall i = 1\ldots n\}$ be a finite set of standard vocabularies defined in the LOM, DC, or vCard standard. Let $NT = \{t_i|\forall i = 1\ldots n\}$ be a finite set of C_{co} and P_{co} terms not defined in the ST; as such, each $t_k \in NT$ is attached to the CO namespace. The set $C_{co} \cup P_{co} = ST \cup NT$.

The example of initial CO structure illustrated in Fig. 2 defines the basic metadata terms (classes and properties) of the e-learning domain consisting of members of the set ST, such as ST = {*dc:title*, *dc:description*, *dc:creator*, *lom:keyword*, etc.}, and the set NT, such as, NT = {*CO:LearningResource*, *CO:Course*, *CO:Unit*, *CO:ResourceFile*, *CO:hasNextCourse*, etc.}.

5. Motivation examples and heterogeneity problems

In most e-learning systems, many learning resource providers provide similar learning resources stored in different database structures and schemata. This section presents examples of two different local ontologies annotated from different learning resource databases extended from our previous work (Banlue, Arch-int, & Arch-int, 2010). These local ontologies express only a portion of the learning resource ontologies provided by learning resource providers who wish to share their learning resources. Examples of such local ontologies, shown in Fig. 3(a) and (b), are called LR1 and LR2 respectively. These two ontologies illustrate the heterogeneity problems occurring in most systems, which are different in both semantic terminology and ontology structure. These local ontologies need to be reconciled by mapping each ontology to the common ontology as a single standard. For purposes of this research, heterogeneity problems were classified into

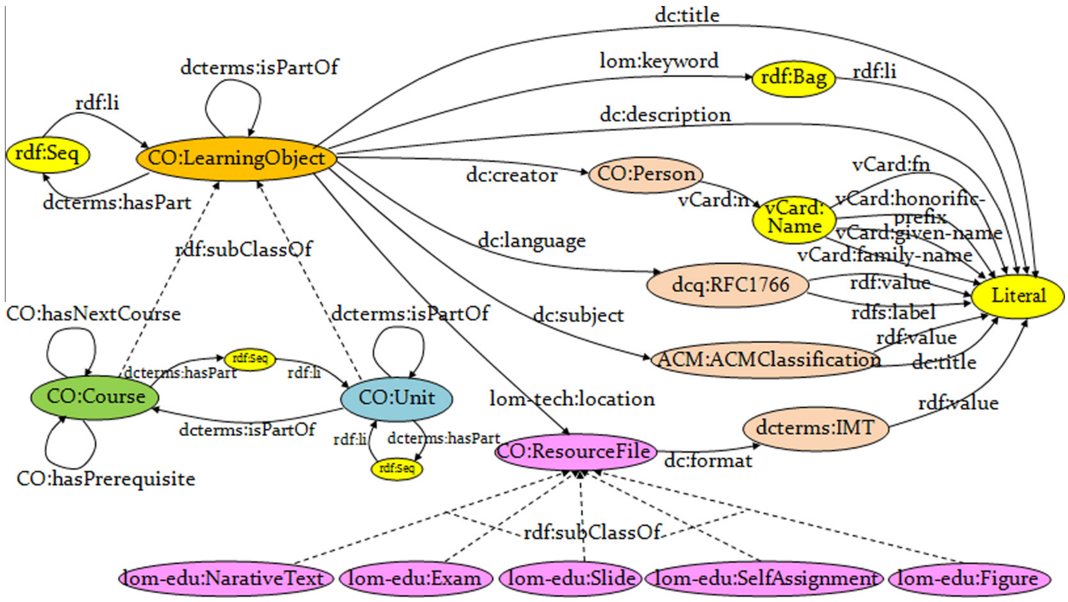


Fig. 2. Common ontology structure.

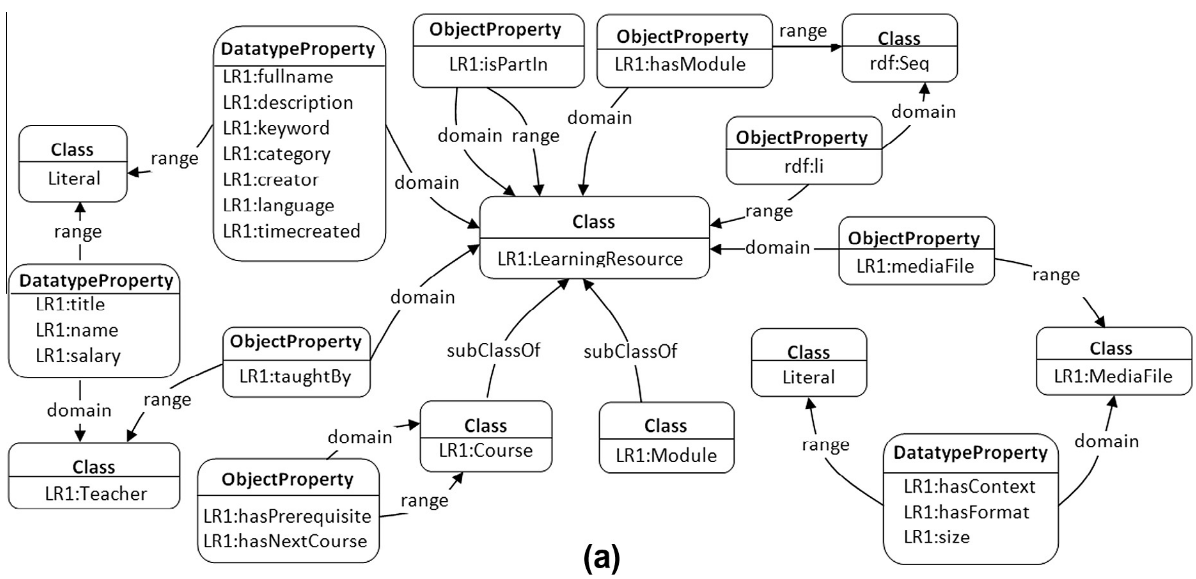


Fig. 3. (a) LR1, a local ontology extracted from the database of learning resource provider 1. (b) LR2, a local ontology extracted from learning resource provider 2.

two main levels, semantic heterogeneity and the structural heterogeneity, as described below.

5.1. Semantic heterogeneity

Semantic heterogeneity occurs when there is a disagreement on the meaning, interpretation, or intended use of the same or related data. Semantic heterogeneity can be classified into various types, such as naming conflict, property-value conflict, scaling conflict, and so on. Due to space limitations, this paper focuses on naming conflict as described below:

- *Naming conflict* encompass two kinds of conflict, synonyms and homonyms. Synonyms define semantically equivalent concepts or properties by different names. For example, the concept *LR1:Module* and the concept *CO:Unit* are synonymous concepts because they both refer to the same fact. The property *LR1:fullname* and the property *dc:title* in CO are also synonymous properties because they both refer to the name of the learning resource. Homonyms, on the other hand, define semantically unrelated concepts or properties by the same name. For example, the property *LR1:title* refers to an academic staff position, whereas the property *dc:title* in *CO* signifies the name of the learning resource.

5.2. Structural heterogeneity

Structural heterogeneity occurs when same concepts or properties are modeled with different logical structures in different systems. In most cases, no direct concept-to-concept or

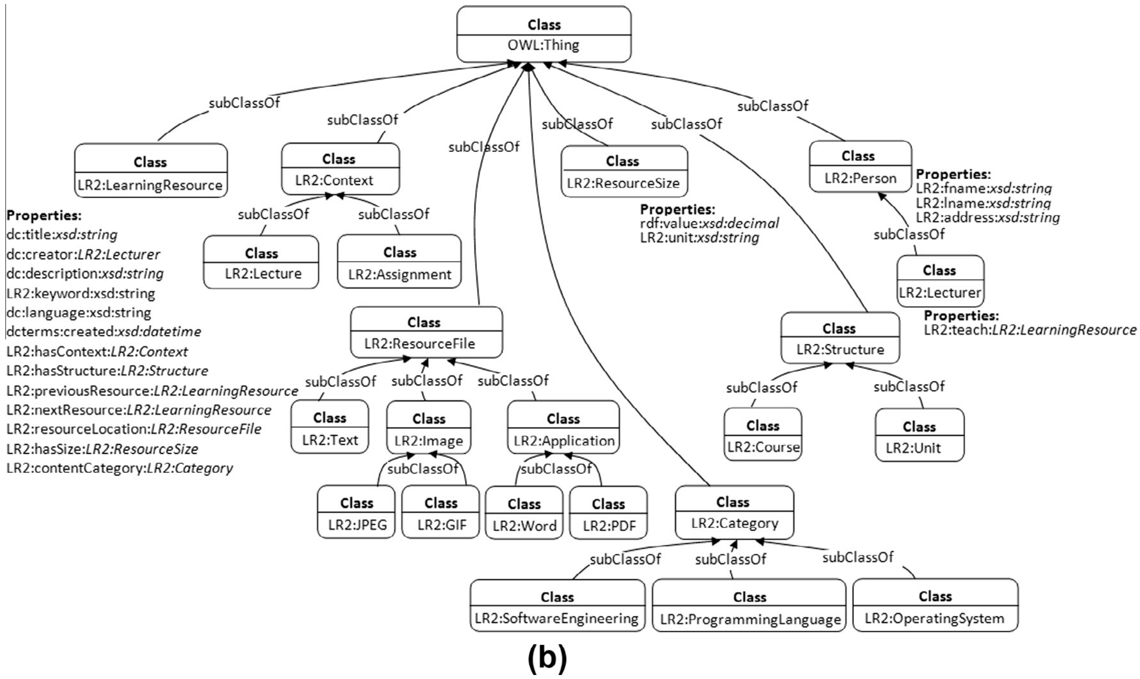


Fig. 3 (*continued*)

Table 1
Examples of structural conflicts.

Conflicts	Description
Generalization conflict	Semantically related concepts are defined in different systems where the concepts in one system subsume the concepts in the other system For example, the concept *CO:Person* in CO subsumes the concept *LR1:Teacher* because the concept *LR1:Teacher* is a subconcept of *CO:Person*;
Isomorphism conflict	Semantically related concepts are described by a dissimilar set of properties For example, the concepts *CO:LearningObject* and *LR1:LearningResource* are semantically related concepts. However, *LR1:Learning Resource* includes the *LR1:size* and *LR1:timecreated* properties, which are not defined in *CO:LearningObject*
Property discrepancies	Semantically equivalent properties from different ontologies have different property types For example, the *LR1:language* property is semantically equivalent to the *dc:language* property, but both properties have different property types: *LR1:language* is a data-type property, whereas *dc:language* is an object property
Concept discrepancies	A concept in an ontology is semantically equivalent to a concept in another ontology which has a property with one specific class or a literal value For example, the *CO:Course* concept in CO is equivalent to the *LR2:LearningResource* concept, whose property *LR2:hasStructure* has its range as the *LR2:Course* concept
Aggregation conflict	A property or a concept in one system maps to a group of properties or concepts, respectively, in another system. For example, the property *LR1:name* of the concept *LR1:Teacher* is equivalent to a group of properties *LR2:firstName* and *LR2:lastName* of the concept *LR2:Person*
Schematic discrepancies	The logical structure of a set of properties and their values belonging to a concept in one ontology are organized to form a different structure in another ontology. This kind of conflict can also be classified several types of conflicts, such as *DataValueProperty conflict*, *PropertyConcept conflict*, and *DataValueConcept conflict* For example, the *DataValueProperty conflict* occurs when the value of a property in one ontology corresponds to a property name in another ontology. The *PropertyConcept conflict* occurs when a property in one ontology is being modeled as a concept in another ontology and the *DataValueConcept conflict* occurs when the value of a property in one ontology corresponds to a concept name in another ontology

property-to-property mapping are possible. Several studies (such as Madnick & Zhu, 2006; Ram & Park, 2004) have classified structural heterogeneity into various types of conflicts. Examples of some conflicts are defined in Table 1.

To facilitate the semantic interoperability of learning resources between systems, these kinds of conflict, both semantic and structural conflicts, must be reconciled using the ontology mapping mechanism. For this paper, we have focused on the resolution of four types of conflicts, i.e., generalization conflict, isomorphism conflict, property discrepancies, and concept discrepancies as described in the following section.

6. The ontology mapping process

Ontology mapping is the process of discovering a corresponding term defined in one ontology which has semantics the same as or similar to a term defined in another ontology. It is a process for mapping different local ontology terminologies onto generic ontology terminologies. For this research, the local ontology vocabulary defined outside of the CO vocabulary must be translated into the CO vocabulary. The result of the ontology mapping process is the generation of the ontology-based learning object metadata (OLOM). The OLOM facilitates user queries across heterogeneous learning resources described using different metadata terminologies. In this research, the Semantic Bridge Ontology is proposed as a core component of the ontology mapping process, as described below.

6.1. The Semantic Bridge Ontology

The proposed Semantic Bridge Ontology (SBO) provides a means to resolve the heterogeneity problems (both semantic and structural) described above. The SBO is designed as a major component to classify different types of conflicts and to provide a systematic way of automatically generating a semantic mapping using a mapping-rule language. The SBO structure enables terms defined in the local ontology to be mapped to terms defined in the common ontology. As a result, the generated mapping rules can transform instances from LROs to instances of CO automatically. Hence, the SBO structure is designed independent of application domains and provides a systematic way to be extended to other kinds of conflict.

The structure of the SBO is expressed in OWL and consists of three main different bridge classes: *EquivalentBridge*, *GeneralizationBridge*, and *SchematicBridge*. *SchematicBridge* can be further classified into three bridge subclasses: *IsomorphismBridge*, *ConceptDiscrepanciesBridge*, and *PropertyDiscrepanciesBridge*. All these bridge classes are defined as subclasses of the *SemanticBridgeOntology* class, as shown in Fig. 4. Each bridge class inherits properties defined in the superclass and also contains its own properties.

These SBO components are used to establish mapping of knowledge between LO and CO, as described in the next subsection.

6.2. Ontology mapping components

To detect various kinds of conflict and to provide a systematic way to map knowledge between LO and CO, this step requires the construction of the ontology mapping components defined as follows.

Definition 9. An ontology mapping, denoted as *OM*, is defined as a tuple:

$$\boldsymbol{OM} = \langle T_{co}, T_{lo}, \phi, \boldsymbol{\omega}, \check{R} \rangle,$$

where $T_{\boldsymbol{co}} = \{C_{co} \cup P_{co}\}$ is a set of CO terms consisting of common concepts and common properties; $T_{\boldsymbol{lo}} = \{C_{lo} \cup P_{lo}\}$ is a set of local ontology terms consisting of local concepts and local properties; Φ is a set of conflict detection rules defined to characterize various kinds of conflict, both semantic and structural conflicts; ω is a set of SBO relationship knowledge defined as a set of triplets and used as a mechanism for generating semantic mapping rules; $\check{R}$ is a set of mapping rules generated from the SBO relationship knowledge. These rules enable semantic mapping for each kind of conflict. The conflict detection rules, the SBO relationship knowledge, and the mapping-rule generation process are defined in the following sections.

6.3. Conflict detection and resolution

6.3.1. Semantic heterogeneity

To resolve naming conflict, this paper focuses on synonyms. Synonyms are often resolved through a terminological mapping process which involves mapping of local concepts or properties defined in a local ontology with the common concepts or properties defined in the CO. The conflict detection and resolution process can be defined as follows:

Conflict Detection Rule 1: This conflict detection step calculates the similarity between concept terms or property terms for each ontology by using the WordNet (Miller, 1995) database to compute the degree of similarity and to suggest a terminology correspondence as defined in the following function.

Sim: $T_1 \times T_2 \to V$, where $T_1 = \{t_{li} \in T_{\boldsymbol{lo}} | \forall\ i = 1 \ldots n\}$, $T_2 = \{t_{cj} \in T_{\boldsymbol{co}} | \forall\ j = 1 \ldots m\}$, and $V = \{v_i | \forall\ i = 1 \ldots n$ and $0 \leqslant v_i \leqslant 1$, with v_i being the similarity value$\}$.

A term $t_{lk} \in T_{\boldsymbol{lo}}$ is mapped onto $t_{ck} \in T_{\boldsymbol{co}}$ if and only if both t_{lk} and t_{ck} are semantically equivalent concepts (or properties), which is denoted by $t_{lk} \cong t_{ck}$ and $v_k \in V$ is equal to 1. In other words, the terms t_{lk} and t_{ck} are in the same synset.

To calculate the similarity value, this research uses the equation proposed by Wu and Palmer (wup) (Wu & Palmer, 1994), which was designed on the basis of the WordNet database. If $Sim_{wup}(t_{lk}, t_{ck}) = 1$, then t_{lk} and t_{ck} are in the same synset, i.e., they have similar meanings, although different words are used. (Details and examples were shown in Banlue et al., 2010).

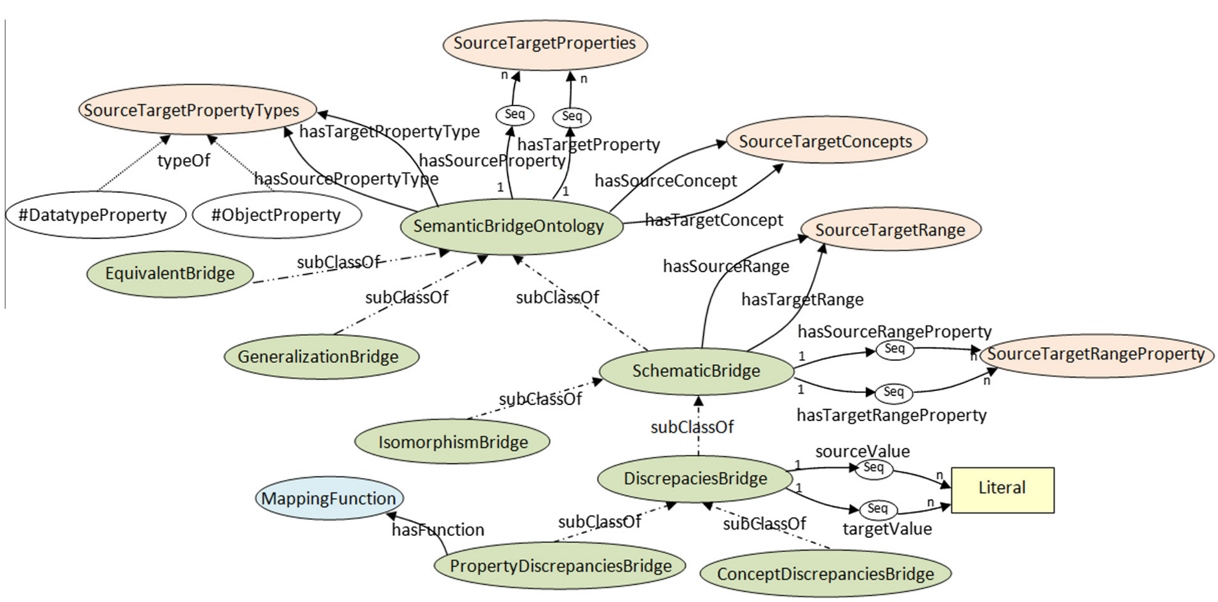


Fig. 4. Semantic Bridge Ontology Structure.

Table 2
EquivalentBridge generation and conflict resolution.

EquivalentBridge	Conflict resolutions
(?eb1, rdf:type, SBO:EquivalentBridge), (?eb1, SBO:hasSourceConcept, ?c_{lk}), (?eb1, SBO:hasTargetConcept, ?c_{ck})	(c_{lk}, equivalentClass, c_{ck})
(?eb2, rdf:type, SBO:EquivalentBridge), (?eb2, SBO:hasSourceProperty, ?p_{lk}), (?eb2, SBO:hasTargetProperty, ?p_{ck})	(p_{lk}, equivalentProperty, p_{ck})

Conflict Resolution for Rule 1: If naming conflict is detected in Rule1, the synonymous terms for concepts (or properties) are related through the *EquivalentBridge* class. For this bridge type, synonymous concepts are related through the *equivalentClass* relationship, whereas synonymous properties are related through the *equivalentProperty* relationship, as illustrated in Table 2.

For example, the pair of *LR1:Module* and *CO:Unit* classes and the pair of *LR1:hasModule* and *dcterms:hasPart* properties are semantically equivalent concepts and properties respectively. These pairs of terms are related through the *EquivalentBridge* class, and conflict resolution is achieved through the mapping rules illustrated in Table 3.

6.3.2. Structural heterogeneity

To resolve structural heterogeneity, this paper focuses on resolving four kinds of conflict, i.e., generalization conflict, isomorphism conflict, property discrepancies, and concept discrepancies, as described below:

(1) Generalization conflict

Conflict Detection Rule 2: A concept in one ontology is defined as a subclass or superclass of a concept of another ontology if these both have the generalization or is-a relationship. To detect this kind of conflict, the function $Sim_{wup}(c_1,c_2)$ can be used. For this research, if $0.7 \leqslant Sim_{wup}(c_1,c_2) < 1$, then it can be inferred that the c_1 and c_2 concepts have the generalization relationship.

Conflict Resolution for Rule 2: If generalization conflict is detected as in Rule 2, the concepts are related through the *GeneralizationBridge* class, as illustrated in Table 4. For this bridge type, the local concept, c_{lk}, must be defined as a concept in the CO, $c_{lk} \in C_{co}$, to enable the user to view the local concept through the CO concept. There are two possible cases for resolving this kind of conflict, as illustrated in Table 4.

For example, the *CO:Person* class has no *subClassOf* relationship in CO, and the *CO:Person* class subsumes the *LR1:Teacher* class. Hence, *COImage*(*LR1:Teacher*, *CO:Teacher*) and (*LR1:Teacher*, *equivalentClass*, *CO:Teacher*), and then (*CO:Teacher*, *subClassOf*, *CO:Person*). As for the LR2 local ontology, the *CO:Person* class subsumes the *LR2:Lecturer class*. Because (*CO:Teacher*, *subClassOf*, *CO:Person*) and Sim_{wup}(*CO:Teacher*, *LR2:Lecturer*) = 1, then (*CO:Teacher*, *equivalentClass*, *LR2:Lecturer*). Therefore, it can be inferred that (*LR1:Teacher*, *subClassOf*, *CO:Person*) and (*LR2:Lecturer*, *subClassOf*, *CO:Person*).

(2) Isomorphism conflict

Conflict Detection Rule 3: If a concept, $c_{lk} \in C_{lo}$, from a local ontology and a concept, $c_{ck} \in C_{co}$, from CO are semantically equivalent concepts, but the concepts have dissimilar sets of properties, then these two concepts have an isomorphism conflict.

Conflict Resolution for Rule 3: Let $\forall p_{li} \in P_{lo}$ be local properties; in other words, $\underline{D}(p_{li}) = c_{lk}$. Let $\forall p_{ci} \in P_{co}$ be common properties, or in other words, $\underline{D}(p_{ci}) = c_{ck}$. Let $\forall v_i \in ST$ be standard vocabularies defined in the LOM, DC, or vCard standard. Two cases must be considered for mapping the dissimilar set of properties for each pair of similar concepts, that is,

I. If $(c_{lk} \cong c_{ck})$ and $\exists p_{lk} \forall p_{ci}$ $(p_{lk} \ncong p_{ci})$, then the local property p_{lk} can be matched with a standard term, $v_c \in ST$. If $(Sim_{wup}(p_{lk},\ v_c) = 1)$, then conflict resolution is performed as in Algorithm I. If the p_{lk} cannot be matched with a standard term, then an image property p_{sm} of the p_{lk} is created, denoted as $COImage(p_{lk},p_{sm})$, and the p_{sm} property is attached to the CO namespace. Then conflict resolution is performed as in Algorithm II. The new v_c or p_{sm} property is then defined as a property of c_{ck} so that users can view and invoke this property from CO. This paper classifies four possible cases for resolving this kind of conflict, as illustrated in Table 5.

Table 3
Examples of *EquivalentBridge* generation and conflict resolution.

EquivalentBridge	Conflict resolution
(?eb1, rdf:type, SBO:EquivalentBridge), (?eb1, SBO:hasSourceConcept, LR1:Module), (?eb1, SBO:hasTargetConcept, CO:Unit)	(LR1:Module rdf:type owl:Class) (CO:Unit rdf:type owl:Class) -> (LR1:Module owl:equivalentClass CO:Unit)
(?eb2, rdf:type, SBO:EquivalentBridge), (?eb2, SBO:hasSourceProperty, LR1:hasModule), (?eb2, SBO:hasTargetProperty, dcterms:hasPart)	(LR1:hasModule rdf:type rdf:Property) (dcterms:hasPart rdf:type rdf:Property) -> (LR1:hasModule owl:equivalentProperty dcterms:hasPart)

Table 4
GeneralizationBridge generation and conflict resolution.

GeneralizationBridge	Conflict resolution
(?gb1, rdf:type, SBO:GeneralizationBridge), (?gb1, SBO:hasSourceConcept, ?c_{lk}), (?gb1, SBO:hasTargetConcept, ?c_{ck})	I. $\forall c_j \in C_{co}$ $(\neg(c_j,\ subClassOf,\ c_{ck})) \Rightarrow COImage(c_{lk},c_{sm})$ and (c_{lk}, *equivalentClass*, c_{sm}) and (c_{sm},*subClassOf*, c_{ck}), where *COImage*(c_{lk},c_{sm}) is defined as a predicate symbol for creating an image concept c_{sm} that represents the same concept of c_{lk}; as such, the c_{sm} concept is attached to the CO namespace II. $\exists c_j \in C_{co}$ $(c_j,\ subClassOf,\ c_{ck})$ *and* $(Sim_{wup}(c_j,c_{lk}) = 1) \Rightarrow (c_j,\ equivalentClass,\ c_{lk})$

Algorithm I: Conflict resolution when the local property (p_{lk}) is matched with the standard property (v_c).

If $\exists v_c$ $Sim_{wup}(p_{lk}, v_c) = 1$ // found the similarity between the local property (p_{lk}) and a standard term ($v_c \in ST$)
Then
  Let $p_{tk} = v_c$; // p_{tk} means the target property
  Let $p_{tk} \in P_{co}$ and $\bar{D}(p_{tk}) = c_{ck}$; // set the target property p_{tk} as a property of CO
  If $\Gamma(p_{lk})$==$\Gamma(p_{tk})$ then // p_{lk} and p_{tk} have the same property type,
Then
    If ($\Gamma(p_{lk})$ = ObjectProperty) and ($\Gamma(p_{tk})$ = ObjectProperty)
  Then
      Let $r_{lk} = \check{R}(p_{lk})$; // set r_{lk} as the range of the p_{lk} property
      Let $r_{ck} = \check{R}(p_{tk})$; // set r_{ck} as the range of the p_{tk} property
    EndIf
    Perform IsomorphismBridgeGeneration as in cases I, II, III of Table 5;
      // generate mapping rules to bridge instances from local ontology to common ontology
  Else
    Perform PropertyDiscrepanciesBridgeGeneration
      // property discrepancies have been detected (as defined in the next section)
  EndIf
EndIf

Algorithm II: Conflict resolution when the local property (p_{lk}) is not matched with the any standard property (v_c).

If $\forall v_i$ $Sim_{wup}(p_{lk}, v_i) \neq 1$ // p_{lk} cannot be matched with any standard terms $\forall v_i \in ST$
Then
  COImage(p_{lk},p_{sm}); // an image property p_{sm} of p_{lk} is created and attached to the CO namespace.
  Let $p_{tk} = p_{sm}$; // p_{tk} is set as the target property
  Let $p_{tk} \in P_{co}$ and $\bar{D}(p_{tk}) = c_{ck}$; // set the target property p_{tk} as a property of CO
  If $\Gamma(p_{lk})$ = ObjectProperty
Then
    Let $r_{lk} = \check{R}(p_{lk})$; // set r_{lk} as the range of the p_{lk} property
    COImage(r_{lk},r_{sm}); // an image concept r_{sm} of r_{lk} is created and attached to the CO namespace.
    Let $r_{ck} = r_{sm}$; // r_{ck} is set as the target range
  EndIf
  Perform IsomorphismBridgeGeneration as cases I and II of Table 5;
EndIf

A case example conforming to Algorithm I can be described as follows:

LR1:LearningResource ≅ *CO:LearningObject*, and the *LR1:LearningResource* class includes the *LR1:timecreated property*, which is not defined in the *CO:LearningObject*. Because *LR1:timecreated* ≅ *dcterms:created*, *dcterms:created* is defined as a property of *CO:LearningObject*, or in other words, $\bar{D}$(*dcterms:created*) = CO:LearningObject. *LR1:timecreated* and *dcterms:created* cannot be set as equivalent properties because LR1:timecreated has a range of Literal, while *dcterms:created* has the range of the standard class dcterms:W3CDTF. This leads to property discrepancies which must be resolved as described in the next section.

A case example conforming to Algorithm II can be described as follows:

LR1:LearningResource contains a *LR1:taughtBy* property that is not defined in *CO:LearningObject*. Because the *LR1:taughtBy* property cannot be transformed into any standard term, *CO:taughtBy* is created as an image of *LR1:taughtBy*, *COImage*(*LR1:taughtBy*, *CO:taughtBy*), and then (*LR1:taughtBy*, *equivalentProperty*, *CO:taughtBy*). The new *CO:taughtBy* is then extended to the *CO:LearningObject*, and the *CO:Teacher* class is set to the range of *CO:taughtBy* property corresponding to the *LR1:taughtBy* property (Note: *CO:Teacher* is the class generated after resolving the *Generalization* conflict).

I. If ($c_{lk} \cong c_{ck}$) and $\exists p_{ck} \forall p_{li}$ ($p_{ck} \ncong p_{li}$), there is no need to extend the p_{ck} property into the c_{lk} concept. Because the c_{lk} concept reveals only the local properties to be searched by users, there is no need to enforce the requirement that the c_{lk} concept contain the same properties as the c_{ck} concept.

For example, *CO:LearningObject* ≅ *LR2:LearningResource*, and the *CO:LearningObject* class contains *dc:language* as a property, whereas *dc:language* is not contained in *LR2:LearningResource*. It is, however, not required to add *dc:language* to *LR2:LearningResource*.

(3) Property discrepancies

Conflict Detection Rule 4: If a property, $p_{lk} \in P_{lo}$, from a local ontology and a property, $p_{ck} \in P_{co}$, from CO are semantically equivalent properties and both p_{lk} and p_{ck} have different property types, i.e., $\bar{D}(p_{lk}) = c_{lk} \in C_{lo}$, $\check{R}(p_{lk})$ = *Literal* and $\bar{D}(p_{ck}) = c_{ck} \in C_{co}$, $\check{R}(p_{ck}) = o_{ck} \in C_{co}$, or on the other hand, $\bar{D}(p_{lk}) = c_{lk} \in C_{lo}$, $\check{R}(p_{lk}) = o_{lk} \in C_{lo}$ and $\bar{D}(p_{ck}) = c_{ck} \in C_{co}$, $\check{R}(p_{ck})$ = *Literal*, these two properties have a property discrepancy.

Conflict Resolution for Rule 4: The solution of this conflict depends on the designated mapping functions, as illustrated in the following two function examples:

- Example of *strContain*() function: The data type property *LR1:language* is semantically equivalent to the object property *dc:language* of the CO, which has the *dcterms:RFC1766* class as its range (see Fig. 5). In this case, if the value of the *LR1:language* property can be matched with the value of the *rdf:label* property of the CO through the *strContain*() function, the corresponding instances of *LR1:LearningResource* can be then transformed into instances of the *CO:LearningObject* class. The *PropertyDiscrepanciesBridge* for this case and the mapping rule generated to resolve this kind of conflict are shown in Table 6.

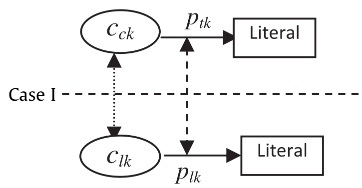
Example of *regex*(*"delimiter"*) function: The data type property *LR1:name* is semantically equivalent to the object property *vCard:n* of the CO, which has the *vCard:Name* class as its range (see Fig. 6). In this case, the value of the *LR1:name* property must be separated into the values of the *vCard:honorific-prefix*, *vCard:given-name*, and *vCard:family-name* properties of the CO through the *regex*(*"delimiter"*) function; the corresponding instance of *LR1:Teacher* can then be transformed into an instance of the *CO:Person* class. The *PropertyDiscrepanciesBridge* for this case and the mapping rule generated to resolve this kind of conflict are shown in Table 7.

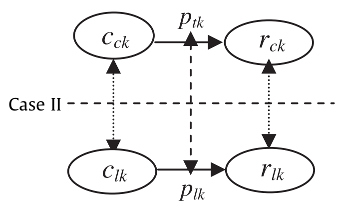
(4) Concept discrepancies

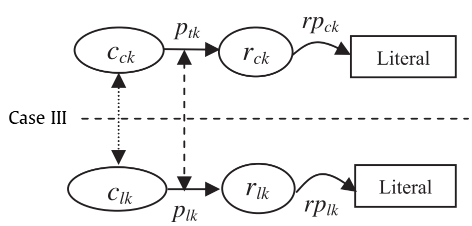
Conflict Detection Rule 5: If a concept, $c_{ck} \in C_{co}$, from CO is semantically equivalent to a concept, $c_{lk} \in C_{lo}$, from a local ontology which has a property p_{lk} whose value is a concept $o_{lk} \in C_{lo}$ or a *Literal*, these two concepts have a concept discrepancy.

Table 5
IsomorphismBridge generation and conflict resolution.

	IsomorphismBridge	Conflict resolution

	(?ib1, rdf:type, IsomorphismBridge),
	(?ib1, hasSourceConcept, ?c_{lk}),
	(?ib1, hasTargetConcept, ?c_{ck}),
	(?ib1, hasSourceProperty, ?p_{lk}),
	(?ib1, hasSourcePropertyType, #DatatypeProperty),
	(?ib1, hasTargetProperty, ?p_{tk}),
	(?ib1, hasTargetPropertyType, #DatatypeProperty)
		IBRule1:
		If ($p_{tk} \in ST$) or ($p_{tk} \notin ST$) Then
		(p_{lk}, equivalentProperty, p_{tk})
		EndIf

	(?ib2, rdf:type, IsomorphismBridge),
	(?ib2, hasSourceConcept, ?c_{lk}),
	(?ib2, hasTargetConcept, ?c_{ck}),
	(?ib2, hasSourceProperty, ?p_{lk}),
	(?ib2, hasSourcePropertyType, #ObjectProperty),
	(?ib2, hasSourceRange, ?r_{lk}),
	(?ib2, hasTargetProperty, ?p_{tk})
	(?ib2, hasTargetPropertyType, #ObjectProperty),
	(?ib2, hasTargetRange, ?r_{ck})
		IBRule2:
		If ($p_{tk} \in ST$) or ($p_{tk} \notin ST$) Then
		(p_{lk}, equivalentProperty, p_{tk}),
		($r_{lk,}$ equivalentClass, r_{ck})
		EndIf

	(?ib3, rdf:type, IsomorphismBridge),
	(?ib3, hasSourceConcept, ?c_{lk}),
	(?ib3, hasTargetConcept, ?c_{ck}),
	(?ib3, hasSourceProperty, ?p_{lk}),
	(?ib3, hasSourcePropertyType, #ObjectProperty),
	(?ib3, hasSourceRange, ?r_{lk}),
	(?ib3, hasSourceRangeProperty, ?rp_{lk}),
	(?ib3, hasTargetProperty, ?p_{tk}),
	(?ib3, hasTargetPropertyType, #ObjectProperty),
	(?ib3, hasTargetRange, ?r_{ck}),
	(?ib3, hasTargetRangeProperty, ?rp_{ck})
		IBRule3:
		If (p_{tk}, r_{ck}, $rp_{ck} \in ST$) Then
		(?x rdf:type c_{lk}),
		(?y rdf:type r_{lk}),
		(?x p_{lk} ?y), (?y rp_{lk} ?z)
		->
		(?x rdf:type c_{ck}),
		(?y rdf:type r_{ck}),
		(?x p_{tk} ?y), (?y rp_{ck}?z)
		EndIf
		Note: in this case, the r_{ck} or r_{lk} may be any anonymous node and may not be set to be an equivalent class directly
		Moreover, rp_{ck} may not be matched with rp_{lk} directly. For example, rp_{ck} represents the *rdf:value* property, whereas rp_{lk} is any property name. However, both rp_{ck} and rp_{lk} contain values with the same meaning

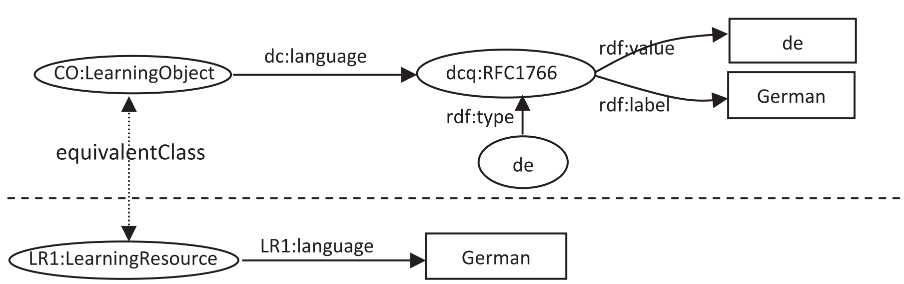


Fig. 5. Example of resolving property type conflict through the *strContain*( ) function.

Examples of such conflicts are presented below:

- Example 1: the *CO:Course* class in CO is semantically equivalent to the *LR2:LearningResource* class which has a property *LR2:hasStructure* with a value from the *LR2:Course* class.
- Example 2: the *lom-edu:NarrativeText* class in CO is semantically equivalent to the *LR1:MediaFile* class which has a property *LR1:hasContext* with a string value of "*Explanation*".

Conflict Resolution for Rule 5: If concept discrepancies are detected as in Rule 5, the *ConceptDiscrepanciesBridge* procedure is initiated; this bridge and the mapping rule generated to resolve conflicts in the above examples are shown in Table 8.

Conflict Detection Rule 6: If a concept, $c_{lk} \in C_{lo}$, from a local ontology is semantically equivalent to a concept, $c_{ck} \in C_{co}$, from CO which has a property p_{ck} whose value is a concept $o_{ck} \in C_{co}$ or a *Literal*, then a concept discrepancy exists. An example of such a conflict is illustrated below:

- The *LR2:JPEG* class from a local ontology is semantically equivalent to the *CO:ResourceFile* class, which has the *dc:format* property whose range is the *dcterms:IMT* class, which has a string value of "*image/jpeg*" (see Fig. 7).

Table 6
PropertyDiscrepanciesBridge generation and conflict resolution.

PropertyDiscrepanciesBridge	Conflict resolution
(?pb1, rdf:type, SBO:PropertyDiscrepanciesBridge), (?pb1, hasSourceProperty, LR1:language), (?pb1, hasTargetProperty, dc:language), (?pb1, hasSourceConcept, LR1:LearningResource), (?pb1, hasTargetConcept, CO:LearningObject), (?pb1, hasSourcePropertyType, #DataTypeProperty), (?pb1, hasTargetPropertyType, #ObjectProperty), (?pb1, hasTargetRange, dcq:RFC1766), (?pb1, hasFunction, strContain()), (?pb1, hasTargetRangeProperty, rdf:label)	Mapping Rule Generation: (?x rdf:type LR1:LearningResource), (?x LR1:language ?l), (?y rdf:type dcq:RFC1766), (?y rdf:label ?c), strContain(?l,?c) ->(?x rdf:type CO:LearningObject), (?x dc:language ?y)

Conflict Resolution for Rule 6: Once concept discrepancies have been detected as in Rule 6, the *ConceptDiscrepanciesBridge* procedure is initiated; this bridge and the mapping rule generated to resolve conflicts for the above examples are shown in Table 9.

6.4. Rule-based reasoning

Once the ontology-based learning object metadata (OLOM) have been generated as a result of mapping rule execution, the OLOM can be augmented with reasoning rules to provide an additional layer of expressivity and to support semantic resource queries. This research used the Jena rule language to define reasoning rules in addition to the mapping rules. These reasoning rules are expressed over the OLOM terminologies to reason about the relationships between individuals of courses or modules and to enhance the semantic query when users need to search for their desired courses or modules. Examples of reasoning rules are presented in RL1–RL6 of Table 10.

7. SMILE system implementation

To demonstrate how the proposed SBO structure could be used in a practical case of conflict resolution and ontology mapping validation, this research developed the SMILE system, which consists of three main components: the SBO mapping tool, the ROM engine, and the semantic query (SQ) engine. The SBO mapping tool was implemented to provide a GUI for the ontology engineer to facilitate SBO creation without using other ontology editor tools. The SBO mapping tool uses the Jena API (Carroll et al., 2004) to provide an interface for importing the LROs and the initial CO. In order to ensure the viability of model realization, the current SMILE system was operated on two Moodle LMSs (http://moodle.org/), which is an open source LMS. These Moodle LMSs were used to store courses in the Computer Science Department, Khon Kaen University, Thailand. The first LMS was developed to store 38 undergraduate courses and another LMS was developed to store 25 graduate courses. Although each LMS has the same database schema and structure, the database schema for each LMS concerned with the learning resources schema were extracted into LROs with different structures as illustrated in the motivation examples of Section 5. In order to obtain learning resource instances, this work exploited Web services (W3C, 2004) to extract data from the database into learning resource instances conforming to the LRO at the schema level. The initial CO was developed by an ontology engineer using ontology editor tools such as Protégé (Stanford., 1999). The JavaFX API (Oracle., 2011) was used to design the visual layout and the graphic interfaces for the ontology mapping procedure between LRO and CO to resolve semantic and structural conflicts. The output of the SBO mapping tool is the SBO generated based on the OWL structure. An example of the SBO mapping tool interface is illustrated in Fig. 8.

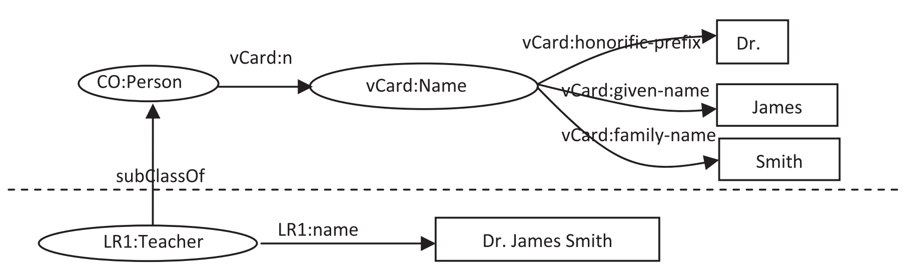


Fig. 6. Example of resolving property type conflict through the *regex*(*"delimiter"*) function.

Table 7
PropertyDiscrepanciesBridge generation and conflict resolution.

PropertyDiscrepanciesBridge	Conflict resolution
(?pb2, rdf:type, SBO:PropertyDiscrepanciesBridge), (?pb2, hasSourceProperty, LR1:name), (?pb2, hasTargetProperty, vCard:n), (?pb2, hasSourceConcept, LR1:Teacher), (?pb2, hasTargetConcept, CO:Person), (?pb2, hasSourcePropertyType, #DataTypeProperty), (?pb2, hasTargetPropertyType, #ObjectProperty) (?pb2, hasTargetRange, vCard:Name), (?pb2, hasFunction, regex(" ")), (?pb2, hasTargetRangeProperty, vCard:honorific-prefix), (?pb2, hasTargetRangeProperty, vCard:given-name), (?pb2, hasTargetRangeProperty, vCard:family-name)	Mapping Rule Generation: (?x rdf:type LR1:Teacher), (?x LR1:name ?n), regex(?n, '(\\w*)\\s*(\\w*)\\s*(\\w*)'), makeTemp(?b1) -> (?x rdf:type CO:Person), (?b1 rdf:type vCard:Name), (?x vCard:n ?b1), (?b1 vCard:honorific-prefix ?n1), (?b1 vCard:given-name ?n2), (?b1 vCard:family-name ?n3) //** function *regex*(" ") receives input ?n = "Dr. James Smith" and separates each field with spaces into ?n1, ?n2, and ?n3 variables, respectively. //** function *makeTemp*() is used to generate a blank node instance (*?b1*) for the target range *vCard:Name*.

Table 8
ConceptDiscrepanciesBridge generation and conflict resolution.

ConceptDiscrepanciesBridge	Conflict resolution
(?cb1, rdf:type, SBO:ConceptDiscrepanciesBridge), (?cb1, hasSourceConcept, LR2:LearningResource), (?cb1, hasTargetConcept, CO:Course), (?cb1, hasSourceProperty, LR2:hasStructure), (?cb1, hasSourcePropertyType, #ObjectProperty), (?cb1, hasSourceRange, LR2:Course)	Mapping Rule Generation: (?*x* rdf:type LR2:LearningResource), (?*x* LR2:hasStructure ?y), (?*y* rdf:type LR2:Course) ->(?*x* rdf:type CO:Course)
(?cb2, rdf:type, SBO:ConceptDiscrepanciesBridge), (?cb2, hasSourceConcept, LR1:MediaFile), (?cb2, hasTargetConcept, lom-edu:NarrativeText), (?cb2, hasSourceProperty, LR1:hasContext), (?cb2, hasSourcePropertyType, #DataTypeProperty), (?cb2, sourceValue, "Explanation")	Mapping Rule Generation: (?x rdf:type LR1:MediaFile), (?x LR2:hasContext "Explanation") ->(?x rdf:type lom-edu:NarrativeText)

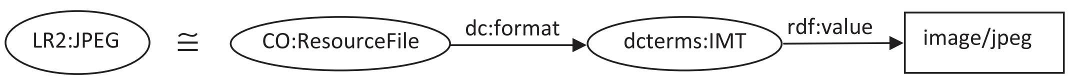


Fig. 7. Concept discrepancies between *LR2:JPEG* from a local ontology and *CO:ResourceFile* from the CO ontology.

Table 9
ConceptDiscrepanciesBridge generation and conflict resolution.

ConceptDiscrepanciesBridge	Conflict resolution
(?cb3, rdf:type, SBO:ConceptDiscrepanciesBridge), (?cb3, hasSourceConcept, LR2:JPEG), (?cb3, hasTargetConcept, CO:ResourceFile), (?cb3, hasTargetProperty, dc:format), (?cb3, hasTargetPropertyType, #ObjectProperty), (?cb3, hasTargetRange, dcterms:IMT), (?cb3, hasTargetRangeProperty, rdf:value), (?cb3, targetValue, "Image/jpeg")	Mapping Rule Generation: (?x rdf:type LR2:JPEG) ->(?x rdf:type CO:ResourceFile), (?x dc:format ?y), (?y rdf:type dcterms:IMT), (?y rdf:value "Image/jpeg")

The SMILE system also used the Jena API and the Jena OWL reasoner to implement the ROM engine and the SQ engine. The imported SBO and LROs and the initial CO are processed by the ROM engine to generate mapping rules automatically. The OLOM is then generated through a mapping-rule translation. In addition, the ROM engine is integrated with the Jena rule engine to provide an interface for augmenting a rich rule-based reasoning system for deducing new knowledge from OLOM knowledge. To facilitate semantic queries, the SMILE system used JavaServer Faces to implement the SQ engine, which provides a server-side user interface to interact with the OLOM to query resources with semantics using SPARQL as a query language. The SQ engine enables other users to query and manipulate (add/update/delete) learning resources from the OLOM, on which users can pose their queries over the CO terminologies which abstract the users from the underlying heterogeneous learning resources. Fig. 9 illustrates the SQ engine interface for users to pose query criteria for searching the desired learning resources. The semantic query results return courses that match the user's conditions, as well as related courses processed using reasoning rules and OWL properties.

8. Contributions and comparison with other approaches

The major contribution of the SMILE system is to propose semantic ontology mapping techniques to resolve the semantic and structural conflicts and enable the interoperability of heterogeneous information systems by applying them to the e-learning domain. We designed the Semantic Bridge Ontology (SBO) as a core component for conflict classification and resolution. We implemented the SBO mapping tool for conflict resolution by providing a mapping interface to bridge local ontology terminologies to common ontology terminologies and auto-generate the SBO structure. The SBO provides a means to auto-generate semantic mapping rules to transform instances from various local ontologies to be instances of the standard common ontology. The structure of SBO was designed on the basis of OWL, which provides a flexible means to extend more for other kinds of conflict. Focusing on OWL-based ontology mapping techniques to enable the interoperability of heterogeneous information systems, the SMILE approach is different from other approaches chosen from the literature review section, as well as our previous works, as described as follows:

In the context of resolving semantic conflicts in ontology mapping, Seng and Kong (2009) provided the structural and systematic intelligent information integration with the use of ontology to enhance both structural and semantic interoperability of the underlying heterogeneous information sources. This work proposes the schema transformation rules to transform the heterogeneous native schemas of the underlying information sources into an XML common data model. To cope with the semantic heterogeneity between different sources, the global schema is designed to connect with the OWL-based ontology by viewing as instances of ontology. This connection aims to enhance the query-processing capability by taking advantage of global schema and query reformulation characteristics. Although, they use ontology to connect with global schema to provide semantic interoperability between different sources, the main drawback of this approach is that they still need to create query reformation and decomposition of global query into native query languages corresponding to each local source. This leads to an overhead at query time and cannot utilize semantic query over the proposed ontology. In addition, the approach indicates integration rules used to cope with semantic and structural heterogeneity of integration processes and also classifies several Kinds of conflict, but still lacks conflicts resolution details. Another approach is that of Liu, Huang, Liu, and Zhong (2007), who introduced a semantic conflict representation model (SCM) based on a semantic conflicts classification framework. The SCM includes two important parts, an ontology (SCO) describing the classification of semantic conflicts and an abstract semantic model (ESM) extending RDB schema. They are described by RDF graph and OWL language respectively. They also propose the algorithms

Table 10
Examples of reasoning rules for enhancing semantic queries.

RuleNo.	Reasoning rule	Description
RL1	(?x rdf:type CO:LearningObject), (?x dc:title ?t), (?y rdf:type CO:LearningObject), (?y lom:keyword ?b), (?b rdf:type rdf:Alt), (?b ?rl ?k), (?r1 rdf:type rdfs:ContainerMembershipProperty), strContain(?t, ?k) ->(?x CO:relatedResource ?y), (CO:relatedResource, rdf:type, OWL:SymmetricProperty)	If there are some modules (or courses) and there exist some keywords of one module (or course) contained in the title of another module (or course), then these modules (or courses) are inferred to be related resources
RL2	(?x rdf:type CO:LearningObject), (?x dc:description ?d), (?y rdf:type CO:LearningObject), (?y lom:keyword ?b), (?b rdf:type rdf:Alt), (?b ?r1 ?k), (?r1 rdf:type rdfs:ContainerMembershipProperty), strContain(?d, ?k) ->(?x CO:relatedResource ?y), (CO:relatedResource, rdf:type, OWL:SymmetricProperty)	If there are some modules (or courses) and there exist some keywords of one module (or course) contained in the description of another module (or course), then these modules (or courses) are inferred to be related resources
RL3	(?x rdf:type CO:LearningObject), (?x dc:subject ?s1), (?s1 rdf:value ?v1), (?y rdf:type CO:LearningObject), (?y dc:subject ?s2), (?s2 rdf:value ?v2), equal(?v1, ?v2) ->(?x CO:relatedResource ?y), (CO:relatedResource, rdf:type, OWL:SymmetricProperty)	If there are some modules (or courses) and these modules (or courses) are in the same category, then these modules (or courses) are inferred to be related resources
RL4	(?u1 rdf:type CO:Unit), (?u2 rdf:type CO:Unit), (?u1 dcterms:hasPart ?b), (?b rdf:type rdf:Seq), (?b ?r1 ?u2), (?r1 rdf:type rdfs:ContainerMembershipProperty) ->(?u1 CO:relatedResource ?u2), (CO:relatedResource, rdf:type, OWL:SymmetricProperty)	If there are some modules which are part of another module, then these modules are inferred to be related modules
RL5	(?c1 rdf:type CO:Course), (?c2 rdf:type CO:Course), (?c1 CO:hasPrerequisite ?c2), ->(?c1 CO:relatedResource ?c2), (CO:relatedResource, rdf:type, OWL:SymmetricProperty)	If there are some courses and each course has the same prerequisite as other courses, then these courses are inferred to be related courses
RL6	(?c1 rdf:type CO:Course), (?c1 dcterms:hasPart ?b1), (?b1 ?r1 ?u1), (?b1 rdf:type rdf:Seq), (?u1 rdf:type CO:Unit), (?r1 rdf:type rdfs:ContainerMembershipProperty), (?c2 rdf:type CO:Course), (?c2 dcterms:hasPart ?b2), (?b2 ?r2 ?u2), (?b2 rdf:type rdf:Seq), (?u2 rdf:type CO:Unit), (?r2 rdf:type rdfs:ContainerMembershipProperty), (?u1 CO:relatedResource ?u2) ->(?c1 CO:relatedResource ?c2), (CO:relatedResource, rdf:type, OWL:SymmetricProperty)	If there are some courses and each course has some modules as part of it, then these modules are related modules, and these courses are inferred to be related courses

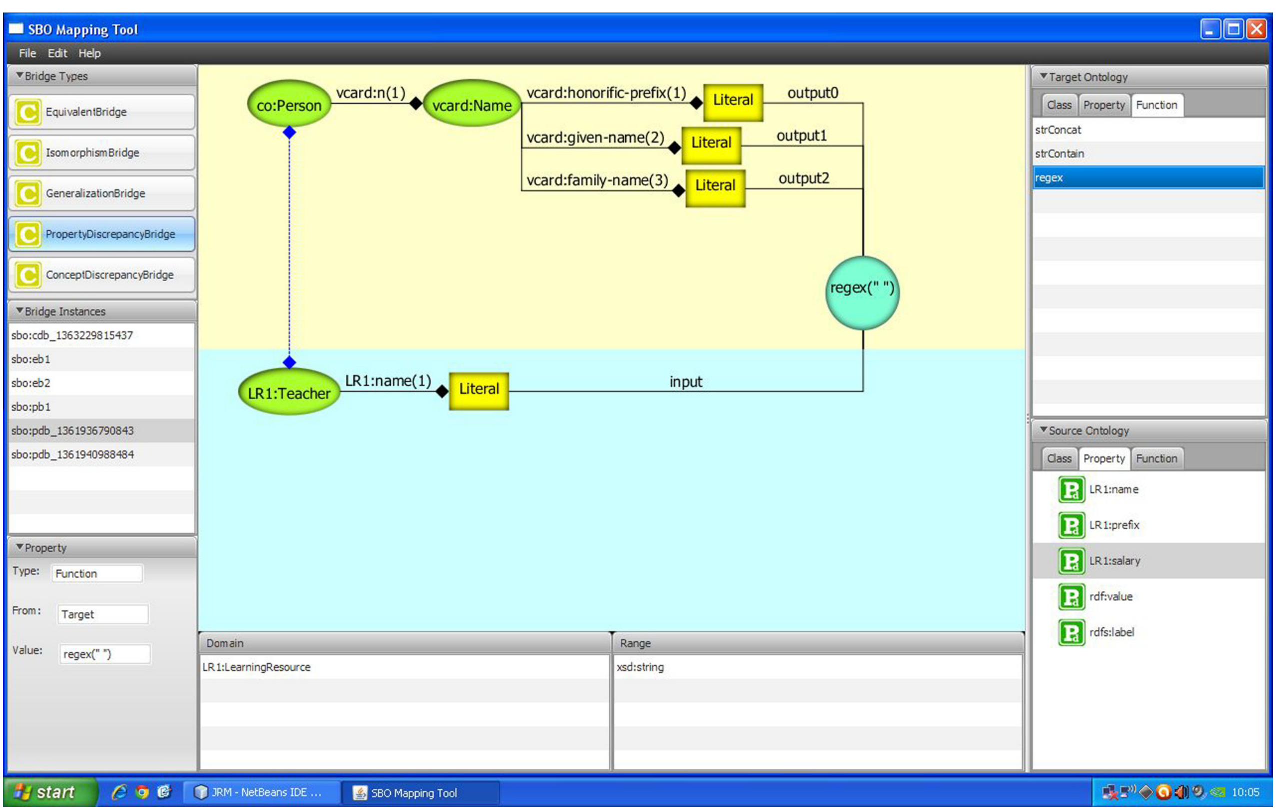


Fig. 8. Example of ontology mapping interface of the SBO mapping tool.

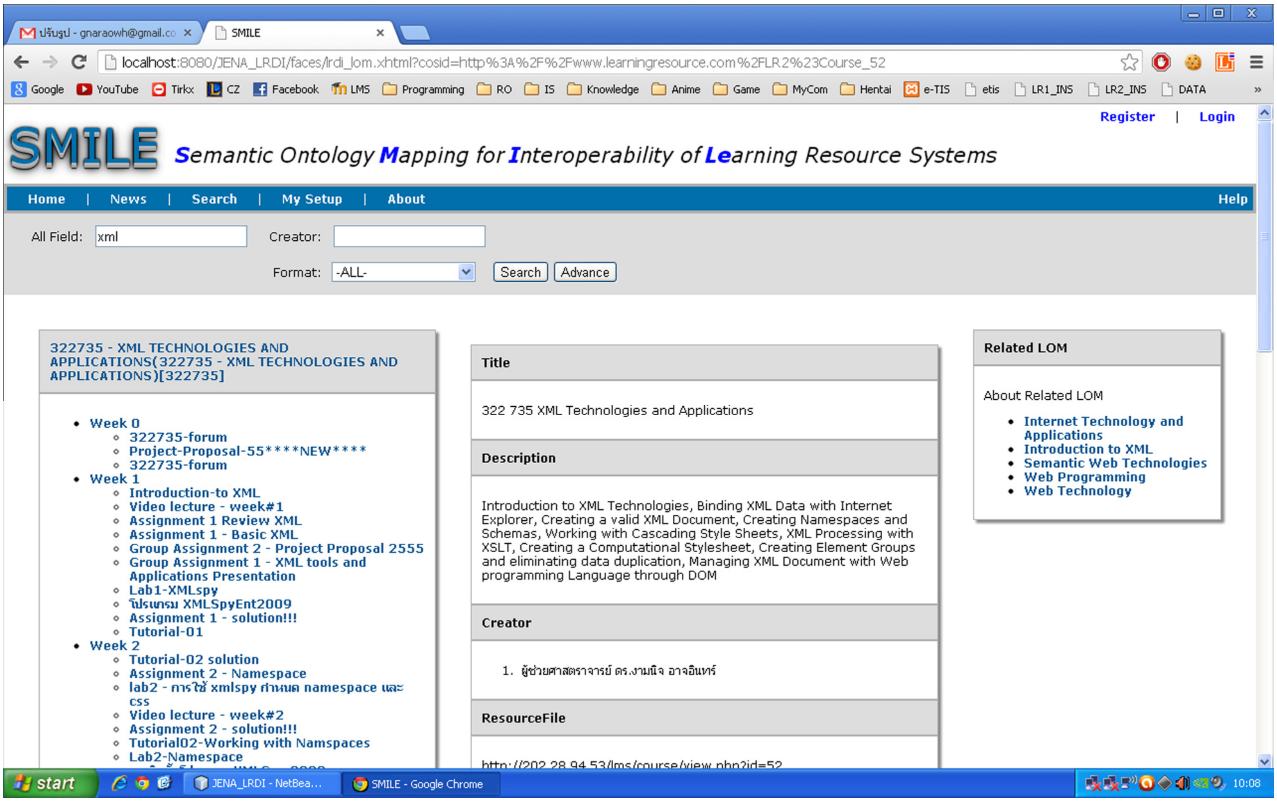


Fig. 9. Example of user query criteria posed in the semantic query interface and the returned results.

based on this model to detect and resolve semantic conflicts. However, this work focuses on solving semantic conflicts at the data level, such as data type conflict, data format conflict, data unit conflict and data precision conflict, but still lacks the schema conflicts detection and resolution. Although the SCO and ESM are designed based on RDF and OWL, they could not exploit the reasoning capability to enhance the conflicts detection and resolution.

In the context of learning objects interoperability, Lee et al. (2011) purposed a multi-strategy knowledge interoperability framework to leverage the interoperability among semantically heterogeneous learning objects and allow users to share their learning materials at a semantic level. They proposed two methodologies, a semantic mapping between knowledge bases (i.e., ontologies), which have essentially similar concepts, and an ontology-based classification algorithm to classify the shareable learning objects from heterogeneous repositories into a local knowledge base by their inner meaning instead of through keyword matching. However, the semantic mapping of this work aims to find only similar concepts by using name similarity comparison, and concept neighborhood comparison techniques. Another work in this context is proposed by Biletskiy et al. (2008), who proposed the RuleML-based common ontology of learning objects (in particular, course outlines) in the Semantic Web and using them to help achieve interoperability between contextually different learning objects and learners. The common ontology must be built through integration of initial ontologies consisting of linking semantically identical and equivalent elements from initial ontologies using articulation ontology. The common ontology facilitates the comparison of course outlines (or their fragments) from two different educational systems. The equivalence of course outlines leads to finding equivalent courses from different university curricula with the goal of their interoperation. Although this work exploits the RuleML, which is a Semantic Web rule interchange, as a common ontology to translate or convert fragments between course outlines of different universities. The approach still lacks the automatic derivation of rules generation to eliminate the semantic conflicts between facts of different ontologies.

As for our previous work (Banlue et al., 2010), we focused on coping with some kinds of conflict, such as the naming conflict, generalization conflict, and aggregation conflict. As for the ontology mapping technique, Banlue, Ngamnij and Somjit proposed one-to-one mapping of local ontologies without exploiting the common ontology model. Hence, the underlying heterogeneous learning resources could not be concealed from the users. Moreover, our previous work defined mapping rules through the description logic, which are translated into OWL primitive and OWL restriction properties to implement mapping rules. However, these mapping rules were limited to some kinds of conflict and could not be extended to support other kinds of conflict, such as property discrepancies and isomorphism conflict.

Our SMILE model differs from existing research in the following three main aspects.

(1) *Kinds of conflict to be solved*
For the conflicts resolution of the OWL-based ontology mapping, most works focus on semantic conflicts resolutions, such as naming conflict and data level conflicts. The other kinds of conflict, such as structural conflicts were rarely mentioned in the literature. The SMILE model focuses on resolving both semantic and structural conflicts, since these conflicts usually exist when multiple ontologies must be interoperated. Compared with our previous work (Banlue et al., 2010), we have extended the resolution of structural conflicts by focusing on four types of conflicts, i.e., generalization conflict, isomorphism conflict, property discrepancies, and concept discrepancies.

(2) *Extensibility of conflict types*
Since several kinds of conflict should be expressed explicitly and resolved effectively, at the same time, the conflicts resolution approach should be flexible enough to keep independent of domains and support extensibility of other conflict types. We introduce the Semantic Bridge Ontology or SBO for conflicts classification and resolution with independent of domains. Even though the current SBO structure still does not support all kinds of conflict, such as the data level conflicts and other structural conflicts, the SBO components represented by OWL provide a flexible means to be extended to other kinds of conflict. Compared with other works (such as Liu et al., 2007), this work is the most related research to our work that provides SCO and ESM ontology for conflicts classification and detection of semantic conflicts found in a heterogeneous database. However, the SCO and ESM ontology structure are designed as support for coping with semantic conflicts at data level conflicts and could not flexible enough to be extended to support structural conflicts resolution.

(3) *Automatic mapping rules generation*
Although there are some pieces of research (such as Biletskiy et al., 2008) that propose the Semantic Web rule language as an alternative approach for semantic conflicts resolution in ontology mapping, their proposed RuleML-based common ontology needed to be created manually without the automatic derivation of rules generation. Compared with the SMILE approach, we provide the SBO mapping tool for users to auto-generate SBO structure that provides a systematic way to generate semantic mapping rules automatically. The mapping rules generation not only reduces the burden of manual mapping rules creation, but also enhances the deductive reasoning capability for conflicts resolution in ontology mapping techniques.

9. Conclusion and future work

This work has presented the SMILE system to support semantic ontology mapping and semantic querying of heterogeneous learning objects. The objectives of semantic ontology mapping were semantic and structural conflict detection and resolution. We have proposed the Semantic Bridge Ontology Mapping tool for facilitating semantic ontology mapping and mapping rules generation. The SMILE system also includes applications for management of learning objects and supports semantic querying of learning resources using ontology-based learning-object metadata terminologies. Since the SMILE system was developed under the real LMSs as described in the implementation section, learning object instances in ontology-based learning-object metadata were obtained from the integration of local ontology instances which are extracted from LMS databases. Hence, learning resources returned from the semantic query engine are resources from physical learning resource systems. This ensures that the proposed approach can be applied to a practical case to validate the viability of model realization and to achieve maximal interoperability across heterogeneous learning-resource systems. There are several other interesting extensions of this research which we are currently exploring.

The research presented in this paper has focused on some kinds of conflict, such as naming conflict, generalization conflict, isomorphism conflict, and property and concept discrepancies. The major challenge in the future is extending the SBO mapping tool to cover other kinds of conflict, such as aggregation conflict and schematic discrepancies, as well as the semantic heterogeneity on the instance level (property value/scaling conflicts). Besides, the mapping processes from local ontology to common ontology are still fully manual mapping. The users need to manage such mapping through the SBO interface tools, even though the SBO mapping tool can generate mapping rules automatically. For future work, we intend to enhance the SBO mapping tool to be a semi-automatic mapping tool, since some conflict resolution, such as naming conflict, can be solved automatically. Moreover, the current SMILE system is operated on only one LMS platform, i.e., the Moodle LMS. There are several LMS platforms including open source LMSs, like Atutor (http://atutor.ca/), Moodle (http://moodle.org/) or Dokeos (http://www.dokeos.com/) and commercial LMSs, like BlackBoard Learning system (http://www.blackboard.com/) or Desire2Learn (http://www.desire2learn.com/). These systems should be according to most e-learning standards and specifications and provided a set of functionalities or educational services. Our current work was limited to learning resources interoperability. We envision that other kinds of educational services can be interoperable among different LMS platforms. Hence, additional efforts will be placed on dynamic interoperability between different LMS platforms. To interoperate these different platforms, we have to cope with system heterogeneity, which includes database heterogeneity and platform heterogeneity. We intend to exploit the Semantic Web services to cope with this heterogeneity and facilitate automated educational service discovery, composition and interoperability.

Although, the current SMILE system aims to interoperate heterogeneous information systems in the e-learning domain, the SBO mapping tool and SBO structure are designed independent of application domains. We are also extending our work to semantic interoperability in the healthcare domain (Sonsilphong & Arch-int, 2013). Since the healthcare information systems or Electronic Patient Records (EPRs) are designed through different standard bodies like HL7, CEN TC251, ISO TC215 and other local standards, it is impossible to easily share data across other healthcare organizations. We intend to apply a SMILE framework to the healthcare domain for integrating patient data stored in heterogeneous EPR systems and supporting health services such as diagnostics and counseling for the treatment by a doctor or specialist. However, we have to cope with information heterogeneity and system heterogeneity. The information heterogeneity, which includes semantic and structural heterogeneities, can be achieved through SBO mapping tool interfaces. To cope with system heterogeneity, we intend to exploit the Semantic Web services to provide a flexible solution for integrating the heterogeneous applications and enabling the dynamic interoperability between different systems. The Semantic Web services can facilitate maximal automation and dynamism in data retrieval to extract data from different databases into ontology instances for each local ontology resource. In addition, The Semantic Web services also provide a means to cope with the semantic conflicts of service descriptions through the Semantic Web services annotation and rule-based inference.

Acknowledgments

This research was granted a monetary support by Khon Kaen University, Thailand, under incubation researcher project, 2010 and the Science Faculty scholarship, 2013. The researchers would like to extend their thanks to Department of Computer Science, Science Faculty, Khon Kaen University, Thailand for the permission to use their venues as the research sites.

References

ADL. (2004). SCORM® (2nd ed.). Overview Retrieved Sep, 01, 2009, from <http://www.adlnet.org>.

Alcaraz Calero, J. M., Marín Pérez, J. M., Bernal Bernabé, J., Garcia Clemente, F. J., Martínez Pérez, G., & Gómez Skarmeta, A. F. (2010). Detection of semantic conflicts in ontology and rule-based information systems. *Data & Knowledge Engineering, 69*(11), 1117–1137. http://dx.doi.org/10.1016/j.datak.2010.07.004.

Banlue, K., Arch-int, N., & Arch-int, S. (2010). Ontology-based metadata integration approach for learning resource interoperability. In *Proceedings of the 6th IEEE international conference on semantics, knowledge & grids* (pp. 195–202). Ningbo, China.

Beydoun, G., Low, G., Tran, N., & Bogg, P. (2011). Development of a peer-to-peer information sharing system using ontologies. *Expert Systems Application, 38*(8), 9352–9364. http://dx.doi.org/10.1016/j.eswa.2011.01.104.

Biletskiy, Y., Boley, H., Ranganathan, R. G., & Boley, H. (2008). RuleML-based learning object interoperability on the Semantic Web. *Interactive Technology and Smart Education, 5*(1), 39–58.

Breitman, K. K., Casanova, A. M., & Truszkowski, W. (2007). *Semantic web: Concepts, technologies and applications*. London: Springer.

Brickley, D., & Guha, R. V. (2004). RDF vocabulary description language 1.0: RDF Schema Retrieved Jan 10, 2006, from <http://www.w3.org/TR/rdf-schema/>.

Bruijn, J. D., Ehrig, M., Feier, C., Martíns-Recuerda, F., Scharffe, F., & Weiten, M. (2006). Ontology mediation, merging, and aligning. In J. Davies, R. Studer, & P. Warren (Eds.), *Semantic web technologies* (pp. 95–113). New York: John Wiley & Sons Ltd.

Brut, M. M., Sedes, F., & Dumitrescu, S. D. (2011). A semantic-oriented approach for organizing and developing annotation for E-learning. *IEEE Transactions on Learning Technologies, 4*(3), 239–248. http://dx.doi.org/10.1109/tlt.2010.40.

Carroll, J. J., Dickinson, I., Dollin, C., Reynolds, D., Seaborne, A., & Wilkinson, K. (2004). Jena: Implementing the semantic web recommendations. In *Paper presented at the proceedings of the 13th international World Wide Web conference on alternate track papers & posters*. New York, NY, USA.

Chen, Y.-J. (2010). Development of a method for ontology-based empirical knowledge representation and reasoning. *Decision Support Systems, 50*(1), 1–20. http://dx.doi.org/10.1016/j.dss.2010.02.010.

Chi, Y.-L. (2009). Ontology-based curriculum content sequencing system with semantic rules. *Expert Systems with Applications, 36*(4), 7838–7847. http://dx.doi.org/10.1016/j.eswa.2008.11.048.

Choi, N., Song, I. I.-Y., & Han, H. (2006). A survey on ontology mapping. *SIGMOD Record, 35*(3), 34–41. http://dx.doi.org/10.1145/1168092.1168097.

Chu, H.-C., Chen, M.-Y., & Chen, Y.-M. (2009). A semantic-based approach to content abstraction and annotation for content management. *Expert Systems with Applications, 36*(2), 2360–2376. http://dx.doi.org/10.1016/j.eswa.2007.12.067.

Chungoora, N., & Young, R. I. M. (2011). Semantic reconciliation across design and manufacturing knowledge models: A logic-based approach. *Applied Ontology, 6*(4), 295–315. http://dx.doi.org/10.3233/ao-2011-0092.

Cuéllar, M. P., Delgado, M., & Pegalajar, M. C. (2011a). A common framework for information sharing in e-learning management systems. *Expert Systems Application, 38*(3), 2260–2270. http://dx.doi.org/10.1016/j.eswa.2010.08.014.

Cuéllar, M. P., Delgado, M., & Pegalajar, M. C. (2011b). Improving learning management through semantic web and social networks in e-learning environments. *Expert Systems with Applications, 38*(4), 4181–4189. http://dx.doi.org/10.1016/j.eswa.2010.09.080.

DCMI. (1990). The Dublin Core Metadata Initiative (DCMI) Retrieved Oct, 20, 2008, from <http://dublincore.org/documents/dces/>.

De Souza, H. C., De, C., Moura, A. M., & Cavalcanti, M. C. (2010). Integrating ontologies based on P2P mappings. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 40*(5), 1071–1082. http://dx.doi.org/10.1109/tsmca.2010.2044880.

Fernández-Breis, J. T., Castellanos-Nieves, D., Hernández-Franco, J., Soler-Segovia, C., Robles-Redondo, M. d. C, González-Martínez, R., et al. (2012). A semantic platform for the management of the educative curriculum. *Expert Systems with Applications, 39*(5), 6011–6019. http://dx.doi.org/10.1016/j.eswa.2011.11.123.

Ferrario, R., Guarino, N., & Prévot, L. (2007). Formal ontologies for communicating agents. *Applied Ontology, 2*(3–4), 209–216.

Gašević, D., & Hatala, M. (2006). Ontology mappings to improve learning resource search. *British Journal of Educational Technology, 37*(3), 375–389. http://dx.doi.org/10.1111/j.1467-8535.2006.00611.x.

Horrocks, I., Patel-Schneider, F. P., Boley, H., Tabet, S., Grosof, B., & Dean, M. (2004). SWRL: A semantic web rule language combining OWL and RuleML Retrieved June, 30, 2009, from <http://www.w3.org/Submission/SWRL/>.

I-Ching, H. (2012). Intelligent discovery for learning objects using semantic web technologies. *Educational Technology & Society, 15*(1), 298–312.

IEEE. (2002). Final 1484.12.1 LOM Draft StandardDocument Retrieved Dec, 12, 2008, from <http://ltsc.ieee.org/wg12/20020612-Final-LOM-Draft.html>.

IMS. (2001). IMS learning resource meta-data information model Retrieved Nov 12, 2008, from <http://www.imsproject.org/>.

Jean-Mary, Y. R., Shironoshita, E. P., & Kabuka, M. R. (2009). Ontology matching with semantic verification. *Web Semantics, 7*(3), 235–251. http://dx.doi.org/10.1016/j.websem.2009.04.001.

Jena. (2010). Reasoners and rule engines: Jena inference support Retrieved Jan, 12, 2011, from <http://jena.apache.org/documentation/inference/index.html>.

Jung, J. J. (2010a). Ontology mapping composition for query transformation on distributed environments. *Expert Systems with Applications, 37*(12), 8401–8405. http://dx.doi.org/10.1016/j.eswa.2010.05.041.

Jung, J. J. (2010b). Reusing ontology mappings for query routing in semantic peer-to-peer environment. *Information Sciences, 180*(17), 3248–3257. http://dx.doi.org/10.1016/j.ins.2010.04.018.

Kalfoglou, Y., & Schorlemmer, M. (2003). Ontology mapping: The state of the art. *Knowledge Engineering Review, 18*(1), 1–31. http://dx.doi.org/10.1017/s0269888903000651.

Kaza, S., & Chen, H. (2008). Evaluating ontology mapping techniques: An experiment in public safety information sharing. *Decision Support Systems, 45*(4), 714–728. http://dx.doi.org/10.1016/j.dss.2007.12.007.

Knight, C., Gašević, D., & Richards, G. (2006). An ontology-based framework for bridging learning design and learning content. *Educational Technology & Society, 9*(1), 23–37.

Lee, M. C., Tsai, K. H., & Hsieh, T. C. (2011). A multi-strategy knowledge interoperability framework for heterogeneous learning objects. *Expert Systems with Applications, 38*(5), 4945–4956. http://dx.doi.org/10.1016/j.eswa.2010.09.157.

Liu, Q., Huang, T., Liu, S., & Zhong, H. (2007). An ontology-based approach for semantic conflict resolution in database integration. *Journal of Computer Science and Technology, 22*(2), 218–227. http://dx.doi.org/10.1007/s11390-007-9028-4.

Madnick, S., & Zhu, H. (2006). Improving data quality through effective use of data semantics. *Data & Knowledge Engineering, 59*(2), 460–475.

Martínez-Gil, J., Delgado-Navas, I., & Aldana-Montes, F. J. (2012). MaF: An ontology matching framework. *Journal of Universal Computer Science, 18*(2), 194–217.

Miles, A., & Bechhofer, S. (2009). SKOS simple knowledge organization system reference Retrieved Nov, 30, 2010, from <http://www.w3.org/TR/skos-reference/>.

Miller, G. A. (1995). WordNet: A lexical database for English. *Communications of the ACM, 38*(11), 39–41. http://dx.doi.org/10.1145/219717.219748.

Muñoz-Merino, J. P., Kloos, D. C., & Naranjo, F. J. (2009). Enabling interoperability for LMS educational services. *Computer Standards Interfaces, 31*(2), 484–498. http://dx.doi.org/10.1016/j.csi.2008.06.009.

Oracle. (2011). JavaFX – The Rich Client Platform Retrieved Oct, 12, 2012, from <http://www.oracle.com/technetwork/java/javafx/overview/index.html>.

Ram, S., & Park, J. (2004). Semantic conflict resolution ontology (SCROL): An ontology for detecting and resolving data and schema-level semantic conflicts. *IEEE Transactions on Knowledge, and Data Engineering, 16*(2), 189–202. http://dx.doi.org/10.1109/tkde.2004.1269597.

Rodriguez, M. A., & Egenhofer, M. J. (2003). Determining semantic similarity among entity classes from different ontologies. *IEEE Transactions on Knowledge and Data Engineering, 15*(2), 442–456. http://dx.doi.org/10.1109/tkde.2003.1185844.

Saake, G., Sattler, K.-U., & Conrad, S. (2005). Rule-based schema matching for ontology-based mediators. *Journal of Applied Logic, 3*(2), 253–270. http://dx.doi.org/10.1016/j.jal.2004.07.021.

Santoso, H. A., Haw, S.-C., & Abdul-Mehdi, Z. T. (2011). Ontology extraction from relational database: Concept hierarchy as background knowledge. *Knowledge-Based Systems, 24*(3), 457–464. http://dx.doi.org/10.1016/j.knosys.2010.11.003.

Schorlemmer, M., & Kalfoglou, Y. (2008). Institutionalising ontology-based semantic integration. *Applied Ontology, 3*(3), 131–150.

Seng, J.-L., & Kong, I. L. (2009). A schema and ontology-aided intelligent information integration. *Expert Systems with Applications, 36*(7), 10538–10550. http://dx.doi.org/10.1016/j.eswa.2009.02.067.

Smith, K. M., Welty, C., & McGuinness, L. D. (2003). Owl: web ontology language guide Retrieved Feb 12, 2006, from <http://www.w3.org/TR/owl-guide/>.

Sonsilphong, S., & Arch-int, N. (2013). Semantic Interoperability for data integration framework using semantic web services and rule-based inference: A case study in healthcare domain. *Journal of Convergence Information Technology (JCIT), 8*(3), 150–159. http://dx.doi.org/10.4156/jcit.vol8.issue3.18.

Stanford. (1999). The protégé ontology editor and knowledge acquisition system Retrieved April, 14, 2009, from <http://protege.stanford.edu/>.

Ullman, D. J. (1988). *Principles of database and knowledge-base systems*. Computer Science Press (Vol. I).

W3C. (2004). Web Services Architecture Retrieved Mar 10, 2007, from <http://www.w3.org/TR/ws-arch/>.

Wu, Z., & Palmer, M. (1994). Verbs semantics and lexical selection. In *Paper presented at the proceedings of the 32nd annual meeting on association for computational linguistics*. Las Cruces, New Mexico.

Yaghmaie, M., & Bahreininejad, A. (2011). A context-aware adaptive learning system using agents. *Expert Systems with Applications, 38*(4), 3280–3286. http://dx.doi.org/10.1016/j.eswa.2010.08.113.

Yu, Z., Zhou, X., & Shu, L. (2010). Towards a semantic infrastructure for context-aware e-learning. *Multimedia Tools and Applications, 47*(1), 71–86. http://dx.doi.org/10.1007/s11042-009-0407-4.