Journal of information and communication convergence engineering
http://jicce.org

J. Inf. Commun. Converg. Eng. 14(2): 97-105, Jun. 2016 Regular paper

Ontology Mapping and Rule-Based Inference for Learning Resource Integration

Kotchakorn Jetinai[1*], Ngamnij Arch-int[2], and Somjit Arch-int[2], *Member, KIICE*

[1]Department of Computer Science, Faculty of Computer Science, Ubon Ratchathani Rajabhat University, Ubon Ratchathani 34000, Thailand
[2]Department of Computer Science, Faculty of Science, Khon Kaen University, Khon Kaen 40002, Thailand

Abstract

With the increasing demand for interoperability among existing learning resource systems in order to enable the sharing of learning resources, such resources need to be annotated with ontologies that use different metadata standards. These different ontologies must be reconciled through ontology mediation, so as to cope with information heterogeneity problems, such as semantic and structural conflicts. In this paper, we propose an ontology-mapping technique using Semantic Web Rule Language (SWRL) to generate semantic mapping rules that integrate learning resources from different systems and that cope with semantic and structural conflicts. Reasoning rules are defined to support a semantic search for heterogeneous learning resources, which are deduced by rule-based inference. Experimental results demonstrate that the proposed approach enables the integration of learning resources originating from multiple sources and helps users to search across heterogeneous learning resource systems.

Index Terms: Conflict resolution, Learning resources, Ontology mapping, Rule-based inference

I. INTRODUCTION

Owing to the rapid growth of e-learning communities, several educational institutions have developed a variety of learning management systems (LMSs) independently, which they have constructed based on their own functional requirements. Therefore, there is a rapidly increasing number of duplicate learning resources (LRs) published on different sites. Metadata plays a crucial role in describing the content of such LRs and in facilitating their integration, to enable the reusability and exchangeability of existing LRs in different LMSs. The use of different types of metadata still results in a problem of interoperability across heterogeneous LRs. Furthermore, most of these metadata standards lack formal semantics and a common standard between heterogeneous metadata descriptions across domains.

Until recently, a number of researchers [1, 2] have used metadata standards and ontologies to semantically annotate LRs; this can easily increase discovery and reuse, and facilitate sharing of LRs among LMSs. Several studies of LR sharing using ontology mapping have been performed [3, 4]. Despite wide acceptance, however, the problems of semantic interoperability and semantic discovery across heterogeneous LMSs still have the potential to cause difficulties in sharing available LRs. The meaning of the information described and the differences in design among information systems lead to *information heterogeneity* problems (or semantic conflicts). In this paper, we classify

Received 11 April 2016, **Revised** 12 April 2016, **Accepted** 28 April 2016
***Corresponding Author** Kotchakorn Jetinai (E-mail: kotchakorn.j@ubru.ac.th, Tel: +66-45-352000 ext.3313)
Department of Computer Science, Faculty of Computer Science, Ubon Ratchathani Rajabhat University, Ubon Ratchathani 34000, Thailand.

Open Access http://dx.doi.org/10.6109/jicce.2016.14.2.097 print ISSN: 2234-8255 online ISSN: 2234-8883

the problem into two main levels, which are described below [5, 6].

1) *Semantic heterogeneity* occurs when there is disagreement about the meaning, interpretation, or intended use of the same or related data.

2) *Structural heterogeneity* occurs when the same concepts are modelled with different logical structures in different systems. In most cases, no direct concept-to-concept mapping is possible.

To address the problems above, we propose a method of resolution for issues of heterogeneity using ontology mapping extended from our previous work [7-9]. We have applied SWRL to solve the problem of ontology mapping, especially in the case of structural conflicts, for which Web Ontology Language (OWL) has a limited capability.

The remainder of the paper is structured as follows: Section II illustrates heterogeneity problems with a motivating example. Section III describes the design of the common ontology that was used in most of this work to overcome the problems outlined above. The ontology mapping for integration is presented in Section IV. Section V presents experimental results obtained via the system implementation and evaluation of the proposed mapping technique. Finally, our conclusions and suggestions for future work are summarized in Section VI.

II. MOTIVATING EXAMPLE

To illustrate the information heterogeneity problems that exist in most e-learning systems, we demonstrate here how two different LR ontologies can extract only a portion of the learning-content resources from distinct LMS repositories. The ontologies are referred to as LR1 and LR2, and are shown in Figs. 1 and 2, respectively.

A. Semantic Heterogeneity

Semantic heterogeneity occurs when there is a disagreement about the meaning, interpretation, or intended use of the same or related data [10, 11]. Semantic heterogeneity is classified into three types: naming conflicts, scaling conflicts, and property value conflicts. These types are described below.

1) Naming conflicts encompass two different kinds of conflict, namely synonyms and homonyms. Synonyms are semantically equivalent concepts or properties defined by different names. For example, the concept (or class) *LR1:Teacher* and the concept *LR2:Lecturer* are synonymous concepts, since they both refer to the same fact. Homonyms, on the other hand, are semantically unrelated concepts or properties defined by the same name. For example, the property *LR1:name* refers to the name of a *Learning Resource*, whereas the property *LR2:name* signifies the name of a *Person*.

2) Scaling conflicts concern semantically equivalent properties defined using different scales (or units of measurement). For example, the properties *LR1:salary* and *LR2:salary* have different units of measurement, ‘EUR’ and ‘USD’, respectively.

3) Property value conflicts concern semantically equivalent properties defined with different property values. For example, the property *LR1:gender* defines ‘M’ and ‘F’ to refer to ‘male’ and ‘female’, whereas the property *LR2:sex* uses ‘0’ and ‘1’ to represent ‘male’ and ‘female’, respectively.

B. Structural Heterogeneity

Structural heterogeneity occurs when the same concepts are modelled with different logical structures in different systems. Although there are several publications that classify structural heterogeneity into various types of conflicts [5, 6, 10, 11], this paper focuses on four main kinds of such conflicts, as described below.

1) Generalization conflicts concern semantically related concepts that are defined in different systems, where the concepts in one system subsume the concepts in another system. For example, the concept *LR1:Members* subsumes the concept *LR2:Guest* since the concept *LR2:Guest* is a subconcept of *LR1:Members*.

2) Aggregation conflicts arise when a property or a concept in one system maps to a group of properties or concepts, respectively, in another system. For example, the property *LR2:name* of the concept *LR2:Person* is equivalent to a group of properties, *LR1:title*, *LR1:firstName*, and *LR1:lastName*, of the concept *LR1:Members*.

3) Property discrepancies concern semantically equivalent properties defined with different property types. For example, the properties *LR1:author* and *dc:creator* in LR2 are semantically equivalent, but the property *LR1:author* is a datatype property, whereas *dc:creator* is an object property.

4) Concept discrepancies occur when the logical structure of a set of properties and their values belonging to a concept in one system are organized to form a different structure in another system. For example, the concept *LR1:Subject* is equivalent to the concept *LR2:Learning Resource*, whose property *LR2:hasStructure* has the concept *LR2:Course* as its range.

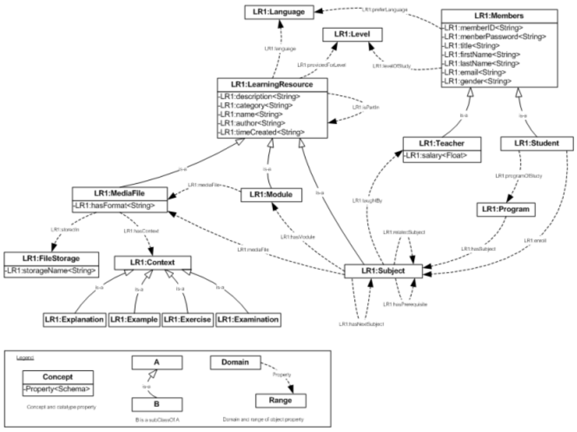

Fig. 1. The source ontology LR1.

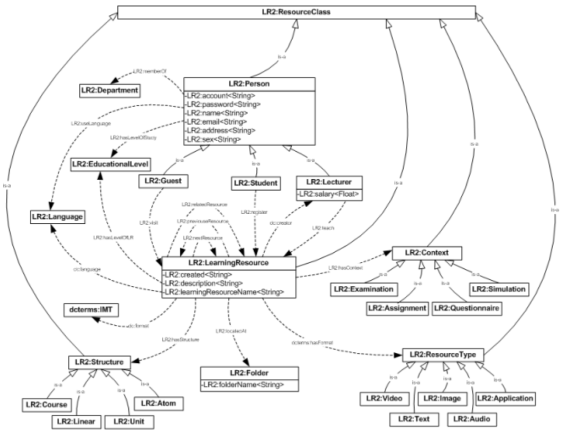

Fig. 2. The source ontology LR2.

III. A COMMON ONTOLOGY

To overcome heterogeneity problems, we designed the Common Ontology (CO) as a standard mediatory ontology for supporting the integration. The CO was defined using basic terms based on standard metadata for e-learning. The two common standards, namely DC and LOM, are included. These standards are aimed at enabling usability, aiding discoverability, and facilitating interoperability, usually in the context of online LMSs. Terminology outside this common vocabulary must be translated to the terminology of the metadata standard; otherwise, a comparison of data semantics will not be possible. In this paper, the common terms that cannot be described with the DC and LOM vocabularies are prefixed with the namespace *CO*, as depicted in Fig. 3.

The CO structure consists of standard concepts and standard properties. The standard concepts include, for example, *CO:LearningResource*, *CO:Course*, *CO:Unit*, *CO:LearningResourceFile*, *dcterms:IMT*, and so on.

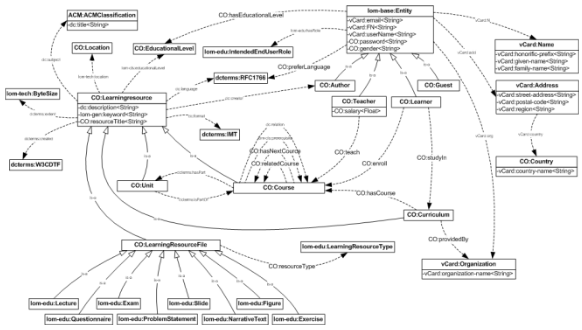

Fig. 3. The common ontology.

The standard properties are the datatype property and the object property. A datatype property, such as *dc:description*, has a literal as its range and defines the datatype using a built-in XML schema. A property which has a concept as its range, such as *dc:format*, is called an object property. The arrows in Fig. 3 labelled ‘is-a’ (*subClassOf*) establish a relationship between concepts in the form of a subsumption hierarchy. Other concepts and properties can be extended into the CO because of its scalability. The standard CO provides an abstract view for users to access information about each local ontology.

IV. ONTOLOGY MAPPING

A. Mapping Rules

This section describes the method of conflict detection and resolution for overcoming the problems mentioned in Section II. The conflict detection method is presented in the form of rules and algorithms, whereas the resolution method is presented as mapping rules, which are used to map between any local ontology and the CO. In our approach, LR1 and LR2 must be mapped into a standard-related entity in the CO. Thus, both ontologies have already conducted sharing of their LRs with the CO. In view of this, after mapping has been performed from LR1 and LR2 to the CO, LR1 can map the semantic entities of LR2 automatically, because it knows about their mapping to a common schema from the CO. This will become increasingly beneficial as more educational content providers begin to use the CO for global sharing of a standard mapping to their repositories.

The following definitions are general notations for the CO components. They are represented on the basis of object-oriented programming and set theory.

DEFINITION 1. *CO* is a set of learning-resource metadata based on the Common Ontology, defined as a tuple $CO = \langle C_{co}, P_{co}, I_{co}, O_{co}, RC_{co}, RP_{co}, \sigma_{co}\rangle$, where C_{co}, P_{co}, I_{co}, O_{co}, RC_{co}, RP_{co}, and σ_{co} are defined as below.

DEFINITION 2. C_{co} is a finite set of common concepts, $C_{co} = \{c_{ci} \mid \forall\, i = 1\ldots n\}$.

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

DEFINITION 4. I_{co} is a set of individuals (or instances) of the common concepts, $I_{co} = \{\check{I}_{ci} \in c_{cj} \mid \forall\, i = 1\ldots n,\ \forall\, j = 1\ldots m\}$.

DEFINITION 5. O_{co} is a set of range concepts or literal values of the common properties, called *objects*, $O_{co} = \{o_{ci} \in I_{co}$ or $o_{ci} = Literal\ value \mid \forall\, i = 1\ldots n\}$.

DEFINITION 6. RC_{co} is a set of mappings from a concept to a concept and is defined as a set of axioms, $RC_{co} = \{r_c(c_{cm}, c_{ck}) \mid c_{cm}, c_{ck} \in C_{co}$ and $r_c \in \acute{R}_c\}$, where $\acute{R}_c = \{subClassOf, disjoint, union, equivalentClass\}$.

DEFINITION 7. RP_{co} is a set of mappings from a property to a property and is defined as a set of axioms, $RP_{co} = \{r_p(p_{cm}, p_{ck}) \mid p_{cm}, p_{ck} \in P_{co}$, and $r_p \in \acute{R}_p\}$, where $\acute{R}_p = \{subPropertyOf, equivalentProperty\}$.

DEFINITION 8. σ_{co} is a set of atoms conforming to the SWRL axioms. Atoms can be formed as $\sigma_{co} = \{C_{co}(x), P_{co}(x, y), sameAs(x, y), differentFrom(x, y), builtIn(r, x,\ldots)\}$, where x, y are either variables, individuals or literal values. An atom $C_{co}(x)$ holds if x is an instance of the concept C_{co}; an atom $P_{co}(x, y)$ holds if x is related to y by the property P_{co}; an atom $sameAs(x, y)$ holds if x is interpreted as the same object as y; an atom $differentFrom(x, y)$ holds if x and y are interpreted as different objects; and $builtIn(r, x,\ldots)$ holds if the built-in relation r holds on the interpretations of the arguments.

The following functions define the domain, range, and type of the common properties.

Đ: $P_{co} \rightarrow C_{co}$ gives the set of domain concepts (C_{co}) of a property $p_{ck} \in P_{co}$.

Ř: $P_{co} \rightarrow O_{co}$ gives the set of range concepts or literals (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 (Ř(p_{ck}) = $o_{ck} \in C_{co}$), then $\Gamma(p_{ck})$ = *ObjectProperty*. If (Ř(p_{ck}) = Literal), then $\Gamma(p_{ck})$ = *DatatypeProperty*.

DEFINITION 9. An ontology mapping, denoted by *OM*, is defined as a tuple $OM = <T_{co}, T_{lo}, \Phi, \check{R}>$, where

$T_{co} = \{C_{co} \cup P_{co}\}$ is a set of CO terms consisting of common concepts and common properties.

$T_{lo} = \{C_{lo} \cup P_{lo}\}$ is a set of local ontology terms consisting of local concepts and local properties.

Φ is a set of rules defined to detect the conflicts of semantics and structure.

$\check{R}$ is a set of mapping rules to enable semantic mapping for each kind of conflict.

The conflict detection rules and the mapping rules are defined in the following sections.

B. Conflict Detection and Resolution

1) Resolution for Semantic Heterogeneity

(a) Naming-conflict resolution: To resolve the synonyms conflict, the resolution procedure applies the WordNet similarity measure [12] of Wu and Palmer [13] to compute the degree of similarity between two terms and to suggest identical terms in the two ontologies based on an accepted threshold specified by the system.

Rule 1: If the similarity score of two terms is equal to 1, then the two terms are equivalent, i.e.

$$Sim\colon\ T_1 \times T_2 \rightarrow S, \text{ where}$$

$T_1 = \{ t_{lk_i} \in T_{lo} \mid \forall\ i = 1\ldots n\}$, $T_2 = \{ t_{lk_j} \in T_{co} \mid \forall\ j = 1...m\}$, and

$S = \{s_i \mid \forall\ i = 1\ldots n$ and $0 \le s_i \le 1$, with s_i being the similarity score$\}$.

A term $t_{lk} \in T_{lo}$ is mapped onto $t_{ck} \in T_{co}$ if and only if both t_{lk} and t_{ck} are semantically equivalent terms, denoted as $t_{lk} \cong t_{ck}$, and then $s_k \in S$ is equal to 1, i.e. $Sim_{wup}(t_{lk}, t_{ck}) = 1$. This means that the terms t_{lk} and t_{ck} are in the same synset.

The mapping rules for resolving the conflict in other cases are as below, where t_{lk} and t_{ck} are the terms involved:

Mapping Rule 1:

Case 1: t_{lk} and t_{ck} are terms for concepts:

$$c_{lk}(?x) \Rightarrow c_{ck}(?x)$$

Case 2: t_{lk} and t_{ck} are terms for properties:

$$p_{lk}(?x,?y) \Rightarrow p_{ck}(?x,?y) \text{ , where}$$

$p_{lk} \equiv p_{ck}$ iff $\Gamma(p_{lk}) \equiv \Gamma(p_{ck})$, Đ($p_{lk}$) $\cong$ Đ(p_{ck}) and Ř(p_{lk}) $\cong$ Ř(p_{ck}).

(b) Scaling-conflict resolution: The property *LR1: salary* can be converted to the same currency unit as the standard property in the CO.

Rule 2: If the two scaling units are in conflict, then the local unit needs to be converted to the standard unit as in the CO.

A local literal value o_{lk} of *LR1:salary* has a scaling unit (such as EUR), i.e. $p_{lk}(?x, ?y)$ = *LR1:salary*(*?x ?unit1*), where *?unit1* refers to the o_{lk} in the local unit. The standard literal value o_{ck} of *CO:salary* has a different unit (such as USD), i.e. $p_{ck}(?x, ?y) = CO{:}salary(?x, ?unit2)$, where *?unit2* refers to the o_{ck} in another unit. In this resolution, we apply the *multiply* built-in function of SWRL to convert o_{lk} to the unit o_{ck} in the mapping rule below.

Mapping Rule 2:

$$c_{lk}(?x) \wedge p_{lk}(?x, ?unit1) \wedge swrlb{:}multiply$$
$$(?unit2, ?const\_rate, ?unit1) \Rightarrow p_{lk'}(?x, ?unit2)$$

The property $p_{lk'}$ is generated to receive *?unit2* of the property p_{lk} executed by the *swrlb:multiply* function, and '*?const_rate*' is the variable representing the exchange rate for converting from *?unit1* to *?unit2*.

The following mapping rule applies Mapping Rule 2 to resolve the scaling conflict occurring in *LR1:salary*, where the 1.23 is the rate exchange between the EUR and USD.

$$LR1{:}Teacher(?x) \wedge LR1{:}salary(?x, ?eur)$$
$$\wedge\, swrlb{:}multiply(?usd, 1.23, ?eur)$$
$$\Rightarrow LR1{:}salaryUSD(?x, ?usd)$$

(c) Property-value conflict resolution: The values of *LR1:gender* and *LR2:sex* can be converted to the same value as the standard property in the CO.

Rule 3: If the values, o_{lk} and o_{ck}, of the equivalent properties are in conflict, then the local value o_{lk} needs to be converted to the standard value o_{ck} as in the CO.

In this resolution, we apply SWRL to convert the o_{lk} of *LR1:gender* = "M" to o_{ck} = "Male" and to convert o_{lk} = "F" to o_{ck} = "Female" in the mapping rule below.

Mapping Rule 3:

$$c_{lk}(?x) \wedge p_{lk}(?x,\ "a") \Rightarrow p_{lk}(?x,\ "b")$$

where "*a*" and "*b*" refer to the literal values of o_{lk} and o_{ck}, respectively.

The resolution of the conflicts in the case study is presented below:

$$LR1{:}Members(?x) \wedge LR1{:}gender(?x,\ "M")$$
$$\Rightarrow LR1{:}gender(?x,\ "Male")$$
$$LR1{:}Members(?x) \wedge LR1{:}gender(?x,\ "F")$$
$$\Rightarrow LR1{:}gender(?x,\ "Female")$$

2) Resolution for Semantic Heterogeneity

(a) Generalization conflict resolution: This solution considers the association between concepts, where each concept is associated with another concept as a superclass (superconcept) or a subclass (subconcept). Two possible cases can be considered, as follows.

Case 1: A standard concept c_{ck} has no subconcepts (it is a single class), and it subsumes a local concept c_{lk}.

Rule 4: If a concept c_{ck} has no subconcepts but it subsumes a concept c_{lk}, then c_{lk} can be assigned as a subclass of c_{ck}.

Mapping Rule 4: The concept c_{lk} must be assigned as a subclass of a concept c_{ck}, $c_{lk} \in c_{ck}$, when c_{ck} has no subconcepts. The conflict resolution is performed by Algorithm 1 below. An image concept c_{sm} of c_{lk} is copied, denoted by *COImageChild*(c_{lk}, c_{sm}), and has the namespace *CO* attached to it. Thus, the new concept c_{sm} is equivalent to c_{lk}, and then the concept c_{sm} is assigned the standard concept c_{ck} as a subclass of (*subClassOf*), so that users can view and invoke this concept from the CO.

Algorithm 1: Generalization conflict resolution when c_{ck} has no subconcepts

```
If ∀c_j ∈ C_co(subClassOf(c_j, c_ck)) then
/* the local concept is a proper subconcept of the standard concept */

Begin
  COImageChild(c_lk, c_sm)
/* create the image concept c_sm of c_lk and attach it to the CO namespace */
   Let c_lk = c_sm;          /* c_lk is set as the target concept */
   Let c_sm ∈ C_co;         /* set target concept to be a concept of C_co */

  equivalentClass(c_lk, c_sm)
/*assign the equivalence between the source and the target concept */
  subClassOf(c_sm, c_ck)
 /* assign the target concept to be a subconcept of C_co */

EndIf
```

Case 2: A standard concept c_{ck} has subconcepts and subsumes a local concept c_{lk}.

Rule 5: If a concept c_{lk} has a semantic relationship as a *subClassOf* a concept c_{ck} that has some subconcepts, then c_{lk} is defined as a *subClassOf* c_{ck} when c_{lk} is equivalent to a subconcept of that c_{ck}.

Mapping Rule 5: If a concept c_{lk} has a semantic relationship as a *subClassOf* a concept c_{ck} that has subconcepts, c_{lk} is defined as a *subClassOf* that c_{ck} when there is an equivalence between c_{lk} and a subconcept of c_{ck}. The conflict resolution is performed after Algorithm 2 below is executed.

Algorithm 2: Generalization conflict resolution when c_{ck} has subconcepts

If $\exists c_j \in C_{co}(subClassOf(c_j, c_{ck}))$ then
/* the standard concept has subconcepts */

Begin
If $\exists c_j\ Sim_{wup}(c_j, c_{lk}) = 1$ then
/* a similarity between c_{lk} and a subconcept of C_{co} has been found */
equivalentClass(c_j, c_{lk})
/* set equivalence between c_{lk} and c_j */
Else
Perform Algorithm 1 to define the subclass association
EndIf

EndIf

After Mapping Rule 5 has been applied, the concept c_{lk} is automatically assigned as a subconcept of the superconcept c_{ck}, which is denoted as *subClassOf*(c_{lk}, c_{ck}). This is inferred from *equivalentClass*(c_j, c_{lk}). If the class description c_{lk} is defined as a *subClassOf* c_{ck}, then the set of individuals of c_{lk} must be a subset of the set of individuals in the class extension of c_{ck}, such that $\breve{I}_{lk} \in c_{ck}$.

***(b) Aggregation conflict resolution*:** When these conflicts occur, a vCard format for standard metadata is used to resolve the problem by allowing the system to define semantic equality.

Rule 6: If a property in one ontology is similar to a group of properties in another ontology, then it can be mapped to that group of properties in the other ontology, as follows.

Since a group of properties *LR1:title*, *LR1:firstName*, and *LR1:lastName* are semantically related to *vCard:FN* in the CO, the group of these properties can be mapped to the *vCard:FN* property in the CO by the *stringConcat* function. The resolution is shown in Mapping Rule 6.

Mapping Rule 6:

$$LR1{:}Members(?x) \wedge LR1{:}title(?x,\ ?n1) \wedge LR1{:}firstName(?x,\ ?n2) \wedge LR1{:}lastName(?x,\ ?n3) \wedge swrlb{:}stringConcat(?fullName, ?n1,\ "\ ", ?n2,\ "\ ", ?n3) \Rightarrow lom\text{-}base{:}Entity(?x) \wedge vCard{:}FN(?x,\ ?fullName)$$

(c) Property discrepancy resolution: To resolve the conflict, the type of the local property needs to be transformed to a standard type.

Rule 7: If two equivalent properties have different property types, then the type of the local property $\Gamma(p_{lk})$ is mapped into a standard type $\Gamma(p_{ck})$ as the equivalent property in the CO.

Mapping Rule 7 resolves the difference between Γ(*LR1:author*) and Γ(*dc:creator*). Here, $\check{R}$(*LR1:author*) = *Literal*, i.e. Γ(*LR1:author*) = *DatatypeProperty*, whereas $\check{R}$(*dc:creator*) = $o_{ck} \in C_{co}$, i.e. Γ(*dc:creator*) = *Object Property*. Therefore, we need to map *LR1:author* into a standard type as *dc:creator*. The resolution of this conflict depends on the designated mapping functions, as illustrated in the following example. The value of *LR1:author* can be separated into three standard properties, *vCard:honorific-prefix*, *vCard:given-name*, and *vCard:family-name*, using the *swrlb:substringBefore* and *swrlb:substringAfter* functions to split the space in the value of *LR1:author*.

Mapping Rule 7:

$$LR1{:}LearningResource(?x) \wedge LR1{:}author(?x,\ ?n) \wedge swrlb{:}substringBefore(?n1,\ ?n,\ "") \wedge swrlb{:}substringAfter(?n2,\ ?n,\ "") \wedge swrlb{:}substringBefore(?n3,\ ?n2,"\ ") \wedge swrlb{:}substringAfter(?n4,\ ?n2,"\ ") \wedge LR1{:}TempName(?b) \Rightarrow CO{:}Author(?x) \wedge vCard{:}N(?x,\ ?b) \wedge vCard{:}honorific\text{-}prefix(?b,\ ?n1) \wedge vCard{:}given\text{-}name(?b,\ ?n3) \wedge vCard{:}family\text{-}name(?b,\ ?n4)$$

Note that the concept *LR1:TempName* is created to receive the instance (*?b*) that holds the three vCard properties. We have *equivalentClass*(*LR1:TempName*, *vCard:Name*), and *equivalentProperty*(*LR1:author*, *dc:creator*) because Γ(*LR1:author*) = *ObjectProperty*.

(d) Concept discrepancy resolution: The resolution procedure applies a value constraint in OWL DL to link a restriction class to either a class description or a data range.

Rule 8: If a set of properties and values of a concept in one system is similar to that of a concept in another system, but they have different structures, then the values of the properties in one ontology can be organized as a set of instances of an identical concept in another ontology using a specific restriction, as follows.

Mapping Rule 8 below is used to map *LR2:Learning Resource*, whose property *LR2:hasStructure* has the value *LR2:course* as its range, into the related concept *CO:Course*.

Mapping Rule 8:

$$LR2{:}hasStructure(?x,\ LR2{:}course) \Rightarrow CO{:}Course(?x)$$

V. EXPERIMENTAL RESULTS

A. System Implementation

In our experiments, we used LRs from the Moodle LMSs of two different universities as a set of instances for testing. One of these data sources originated from the Department of Computer Science, Khon Kaen University, Thailand, and the other from the Faculty of Computer Science, Ubon Ratchathani Rajabhat University, Thailand. The first source contained six undergraduate and six graduate courses, while the second contained eight undergraduate and three graduate courses. We implemented a system called 'OWLGenerator' to extract database tuples from the LMSs into the ontological instances. The first source was mapped to LR1, and the second source was mapped to LR2. The instances of each local ontology were integrated into the CO through the ontology-mapping process.

To demonstrate how the rules resolved problems, a system named the Learning Resource Integration System (LRIS) was developed to evaluate the proposed approach. The system provided basic and advanced search capabilities. It allowed users to search for LRs and retrieved the results based on the inference rules. We used the advanced search module of the LRIS to execute SPARQL commands to query the LRs across heterogeneous ontologies. The query was to select the LRs that contained the keyword 'program' in the resource title, as shown in the following:

```
SELECT *
WHERE { ?LR CO:resourceTitle ?LRname.
        FILTER REGEX(?LRname, "program", "i").
        ?LR dc:format ?Format.
        ?LR CO:resourceType ?Type.
        ?LR lom-cls:educationalLevel ?Level.
        ?LR dc:language ?Language      }
```

Fig. 4 depicts an excerpt from the list of LRs and properties retrieved from the LR1 and LR2 sources. The results were inferred by use of Mapping Rules 1 and 3 and represent the LR properties corresponding to the query.

B. System Evaluation

To validate the system, three techniques which are considered standard information retrieval metrics [14], namely precision, recall, and F-measure, can be used to evaluate the results of different approaches to information retrieval; they are widely accepted in the evaluation of ontology mapping [15]. The F-measure is a combination of the precision and recall measures. The formulas are as below:

Query Result : 38 answers

No.	LR	LRname	Format	Type	Level	Language
1	LR1XML_CH4	XML_CH4_The_XSLT_Programming_Language.jpg	dcterms:Image_jpeg	LR1:example	CO:Bachelor	dcterms:en
2	LR1WS00	Introduction_to_Web_Services_program.png	dcterms:Image_png	LR1:example	CO:Bachelor	dcterms:en
3	LR1WS02	Create_Web_Services_programming2_J2EE_SDK.bmp	dcterms:Image_bmp	LR1:example	CO:Bachelor	dcterms:th
4	LR2FP_handout006	Searching_and_program.pdf	dcterms:Application_pdf	LR2:assignment	CO:Bachelor	dcterms:en
5	LR2Topic7_CPL	Subprograms.ppt	dcterms:Application_ppt	lom-edu:slide	CO:Bachelor	dcterms:en
6	LR2Java_CH2	The Java Programming Environment.ppt	dcterms:Application_ppt	lom-edu:slide	CO:Bachelor	dcterms:th
7	LR1WS06	Programming_for_Web_Services-WSDL.jpg	dcterms:Image_jpeg	LR1:example	LR1:Master	dcterms:th
8	LR2CPL_handout_topic1	Brief_History_Programming_Languages.bmp	dcterms:Image_bmp	LR2:examination	CO:Bachelor	dcterms:en
9	LR1Handout_018	Handout_network_introduction_and_socket_programming.ppt	dcterms:Application_ppt	lom-edu:slide	CO:Bachelor	dcterms:en
10	LR1WS04	Describing_Web_Services-WSDL_program.jpeg	dcterms:Image_jpeg	LR1:example	CO:Bachelor	dcterms:en
11	LR1XML_CH7	XML_CH7_Managing_multiple_XML_documents_with_XSLT_program.jpg	dcterms:Image_jpeg	LR1:example	CO:Bachelor	dcterms:en
12	LR1handout6_algo	handout6_dynamic_programming.doc	dcterms:Application_msword	lom-edu:narrativeText	LR1:Master	dcterms:en
13	LR2CPL_handout_topic5	Object-oriented_programming_Exception-handling_Java.ppt	dcterms:Application_ppt	lom-edu:slide	CO:Bachelor	dcterms:th
14	LR2Chapter4_CA	Dynamic Programming.ppt	dcterms:Application_ppt	lom-edu:slide	CO:Bachelor	dcterms:th
15	LR2Topic1_CPL	Brief History of Programming Languages.ppt	dcterms:Application_ppt	lom-edu:slide	CO:Bachelor	dcterms:th
16	LR1c322304_Servlet	c322304_Servlet_ASP_programming.png	dcterms:Image_png	LR1:example	LR1:Master	dcterms:en
17	LR2JavaLecture3	Fundamental_Programming_Structures_Java.ppt	dcterms:Application_ppt	lom-edu:slide	CO:Bachelor	dcterms:th
18	LR2LC4122502	Web Programming.html	dcterms:Text_htm	lom-edu:questionnaire	LR1:Master	dcterms:th
19	LR2LC4122207	Java Programming.html	dcterms:Text_htm	LR2:assignment	CO:Bachelor	dcterms:en
20	LR1XML_CH01	XML_CH01_XSLT_and_XPath_Introduction_for_programming.html	dcterms:Text_htm	lom-edu:narrativeText	CO:Bachelor	dcterms:en
21	LR1XML_CH3	XML_CH3_Writing_and_Applying_XSLT_Style_Sheets_programming.bmp	dcterms:Image_bmp	LR1:example	CO:Bachelor	dcterms:en
22	LR2Cont_EC_doc3	B2C_E-Commerce_programming.ppt	dcterms:Application_ppt	lom-edu:slide	CO:Bachelor	dcterms:th
23	LR2Topic8_CPL	Implementing Subprograms.ppt	dcterms:Application_ppt	lom-edu:slide	CO:Bachelor	dcterms:th
24	LR1WS01	Create_Web_Services_programming_by_J2EE_SDK.jpeg	dcterms:Image_jpeg	LR1:example	CO:Bachelor	dcterms:th
25	LR2CPL_handout_topic7	Program_and_Sub_program.ppt	dcterms:Application_ppt	lom-edu:slide	CO:Bachelor	dcterms:th
26	LR2Topic5_CPL	Object-oriented programming and Exception-handling in Java.pdf	dcterms:Application_pdf	LR2:examination	CO:Bachelor	dcterms:th
27	LR2FP_handout007	Sorting_and_program_example.ppt	dcterms:Application_ppt	lom-edu:slide	CO:Bachelor	dcterms:th
28	LR1Handout_017	Handout_network_introduction_and_socket_programming.pdf	dcterms:Application_pdf	lom-edu:narrativeText	LR1:Master	dcterms:en
29	LR1WS05	Describing_Web_Services-WSDL2_programming.png	dcterms:Image_png	LR1:example	CO:Bachelor	dcterms:th
30	LR1XML_CH6	XML_CH6_Programming_for_Mapping_XML_dialects_with_XSLT.bmp	dcterms:Image_bmp	LR1:example	CO:Bachelor	dcterms:en
31	LR2JavaLecture2	Java_Programming_Environment.ppt	dcterms:Application_ppt	lom-edu:slide	CO:Bachelor	dcterms:th
32	LR2CPL_handout_topic8	Implementing_Subprograms.bmp	dcterms:Image_bmp	lom-edu:figure	CO:Bachelor	dcterms:en
33	LR1WS03	Describing_Web_Services_WSDL_programming.jpeg	dcterms:Image_jpeg	lom-edu:figure	CO:Bachelor	dcterms:en
34	LR2CA_onlineresource4	dynamic_programming.png	dcterms:Image_png	lom-edu:figure	LR1:Master	dcterms:th
35	LR1c322941_MICRO	c322941_MICRO_PROGRAMMED_CONTROL.ppt	dcterms:Application_ppt	lom-edu:slide	LR1:Master	LR2:French
36	LR2LC4122306	Computer Programming Language.ppt	dcterms:Application_ppt	lom-edu:slide	CO:Bachelor	dcterms:th
37	LR1c322304_Client	c322304_Client_Side_Programming.bmp	dcterms:Image_bmp	LR1:example	CO:Bachelor	dcterms:en

Fig. 4. LRs and properties retrieved from the LR1 and LR2 sources.

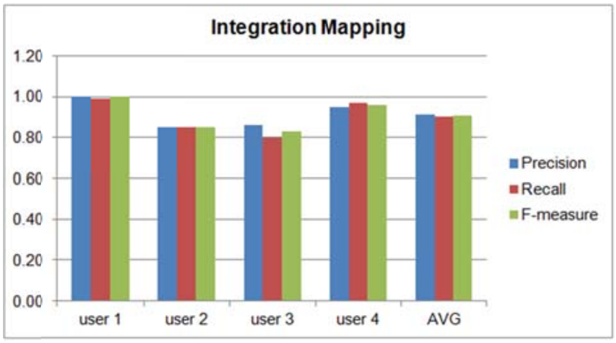

Fig. 5. Results of the evaluation of integration mapping.

$$\text{Precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|}, \quad (1)$$

$$\text{Recall} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|}, \quad (2)$$

$$\text{F} - \text{measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (3)$$

The LRIS was tested by four service requesters (users), i.e. authors, teachers, learners, and guests, who searched for LRs across different LMSs. The results retrieved were then evaluated using the three metrics. Each user executed 30 queries to retrieve the results. The results are shown in Fig. 5.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed a methodology for the integration of LRs for semantic search using ontology mapping and rule-based inference. The proposed approach focuses mainly on addressing the problem of heterogeneity, including both semantic and structural conflicts. The standard CO was designed for a global shared ontology for overcoming the heterogeneity problems among different LMSs. Reasoning rules to cope with the problems were defined using SWRL. The LRIS application for LR discovery was developed in order to evaluate the proposed approach. Experimental results have shown that our proposed approach performs effectively. Moreover, the proposed rules also allow other approaches to be incorporated, despite the existence of distinct platforms and data heterogeneities. The application of a metadata-based ontology enables us to achieve a higher level of interoperability and greater practicability in e-learning domains.

However, the ability of our approach to resolve more complex problems is still limited. The mapping process is not fully automatic. Some conflicts require a domain expert to detect and resolve them manually. Modelling of mediator ontologies has been investigated mainly in the context of its application to e-learning, and does not cover other domains. In future work, we intend to improve the mediator ontology in order to support different educational systems. The semi-automatic mapping tool must be enhanced to support the mapping process. Moreover, more conflict resolution needs to be performed in future work.

ACKNOWLEDGMENTS

This research was supported by Ubon Ratchathani Rajabhat University, Thailand. The authors would like to thank the Department of Computer Science, Khon Kaen University, and the Faculty of Computer Science, Ubon Ratchathani Rajabhat University, Thailand for permission to use their data sets and research equipment.

REFERENCES

[1] M. M. Brut, F. Sedes, and S. D. Dumitrescu, "A semantic-oriented approach for organizing and developing annotation for e-learning," *IEEE Transactions on Learning Technologies*, vol. 4, no. 3, pp. 239-248, 2011.

[2] M. P. Cuellar, M. Delgado, and M. C. Pegalajar, "A common framework for information sharing in e-learning management systems," *Expert Systems with Applications*, vol. 38, no. 3, pp. 2260-2270, 2011.

[3] D. Gasevic and M. Hatala, "Ontology mappings to improve learning resource search," *British Journal of Educational Technology*, vol. 37, no. 3, pp. 375-389, 2006.

[4] A. Bouzeghoub and A. Elbyed, "Ontology mapping for web-based educational systems interoperability," *Interoperability in Business Information Systems*, vol. 1, no. 1, pp. 73-84, 2006.

[5] S. Ram and J. Park, "Semantic conflict resolution ontology (SCROL): an ontology for detecting and resolving data and schema-level semantic conflicts," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 2, pp. 189-202, 2004.

[6] H. Lu and Q. Z. Li, "Ontology based resolution of semantic conflicts in information integration," *Wuhan University Journal of Natural Sciences*, vol. 9, no. 5, pp. 606-610, 2004.

[7] K. Banlue, N. Arch-int, and S. Arch-int, "Ontology-based metadata integration approach for learning resource inter-operability," in *Proceeding of the 6th International Conference on Semantics Knowledge and Grid*, Ningbo, China, pp. 195-202, 2010.

[8] N. Arch-int and S. Arch-int, "Semantic ontology mapping for interoperability of learning resource systems using a rule-based reasoning approach," *Expert Systems with Applications*, vol. 40, no. 18, pp. 7428-7443, 2013.

[9] K. Jetinai, N. Arch-int, W. Rungworawut, and S. Arch-int, "Ontology reconciliation for learning resource interoperability," *International Journal of Digital Content Technology & Its Applications*, vol. 7, no. 6, pp. 191-200, 2013.

[10] C. F. Naiman and A. M. Ouksel, "A classification of semantic conflicts in heterogeneous database systems," *Journal of Organizational Computing*, vol. 5, no. 2, pp. 167-193, 1995.

[11] V. Kashyap and A. Sheth, "Semantic and schematic similarities between database objects: a context-based approach," *The VLDB Journal*, vol. 5, no. 4, pp. 276-304, 1996.

[12] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller, "Introduction to WordNet: an on-line lexical database," *International Journal of Lexicography*, vol. 3, no. 4, pp. 235-244, 1990.

[13] Z. Wu, and M. Palmer, "Verbs semantics and lexical selection," in *Proceeding of the 32nd Annual Meeting on Association for Computational Linguistics*, Las Cruces, NM, pp. 133-138, 1994.

[14] J. Euzenat, "Semantic precision and recall for ontology alignment evaluation," in *Proceeding of the 20th International Joint Conference on Artificial Intelligence*, Hyderabad, India, pp. 348-353, 2007.

[15] S. Kaza and H. Chen, "Evaluating ontology mapping techniques: an experiment in public safety information sharing," *Decision Support Systems*, vol. 45, no. 4, pp. 714-728, 2008.

Kotchakorn Jetinai
received her Ph.D. in computer science from Khon Kaen University, Thailand, in 2014. She is currently an assistant professor at the Faculty of Computer Science, Ubon Ratchathani Rajabhat University. Her current research interests are the semantic web, ontology mapping, information integration, and personalized e-learning.

Ngamnij Arch-int
received her Ph.D. in computer science from Chulalongkorn University, Thailand, in 2003. She is currently an associate professor in the Department of Computer Science at Khon Kaen University, Thailand. Her research interests include the semantic web, web services, semantic web services, and heterogeneous information integration.

Somjit Arch-int
received his Ph.D. in computer science from the Asian Institute of Technology, Thailand, in 2002. He is currently an associate professor in the Department of Computer Science, Khon Kaen University, Thailand. His previous experience includes the development of several industry systems and consulting activities. His research interests are business component-based software development, object-oriented metrics, ERP/Logistics and supply chain, traceability systems, and the semantic web.