

University of Wollongong

## Research Online

---

Faculty of Engineering and Information  
Sciences - Papers: Part A

Faculty of Engineering and Information  
Sciences

---

1-1-2012

### An ontology development methodology to integrate existing ontologies in an ontology development process

Nelson K. Y Leung  
*RMIT University*

Sim Kim Lau  
*University of Wollongong, simlau@uow.edu.au*

Nicole Tsang  
*RMIT University*

Follow this and additional works at: <https://ro.uow.edu.au/eispapers>



Part of the [Engineering Commons](#), and the [Science and Technology Studies Commons](#)

---

Research Online is the open access institutional repository for the University of Wollongong. For further information contact the UOW Library: [research-pubs@uow.edu.au](mailto:research-pubs@uow.edu.au)

---

# An ontology development methodology to integrate existing ontologies in an ontology development process

## Abstract

Ontology is defined as an explicit specification of a conceptualization while a conceptualization is an abstract, simplified view of the world that we wish to represent for some purpose. To build high quality ontologies, developers are required to choose and follow a suitable ontology development methodology in which containing a series of steps, activities and guidelines that are put together in an organized and systematic manner. Literatures show that building ontologies by reusing existing ontologies is more cost effective than building from scratch. However, majority of the methodologies only provide a very limited discussion about how to perform integration or ontology reused in their ontology development processes. This paper aims to develop a set of guidelines for the selection of ontology reuse methods which has not been explored in any ontology research. In addition, an ontology development methodology that incorporates ontology reuse methods and a system that can assist to perform integration semi-automatically is proposed. An application scenario has been developed to illustrate how the proposed development methodology can be adopted to create an ontology.

## Keywords

existing, ontology, ontologies, development, process, methodology, integrate

## Disciplines

Engineering | Science and Technology Studies

## Publication Details

Leung, N. K. Y., Lau, S. Kim. & Tsang, N. (2012). An ontology development methodology to integrate existing ontologies in an ontology development process. *Communications of the ICISA: An International Journal*, 13 (2), 31-61.

VOLUME 13

NUMBER 2

Winter 2012

# COMMUNICATIONS OF THE ICISA

A N I N T E R N A T I O N A L J O U R N A L

## *Contents:*

### **Learning IT Sustainability: A Balanced Scorecard Approach**

*Anne-Marie Guidy-Oulai, Beth Hoger, J. Michael Tarn*

### **Managing Software Evolution Life Cycle: A Demand Perspective**

*Hsiang-Jui Kung, LeeAnn Kung*

### **An Ontology Development Methodology to Integrate Existing Ontologies in an Ontology Development Process**

*Nelson K. Y. Leung, Sim Kim Lau, Nicole Tsang*

### **Accountants' Cognition and Attitude toward the Environmental Accounting System in Taiwan Manufacturing Industry**

*She-I Chang, David C. Yen, I-Cheng Chang, Hsing-Jung Li, Su-Han Cheng, Lin-Wan Chen*

### **Applying RFM Analysis to Library-Book Recommender Systems**

*Ya-Han Hu, Chia-Lun Lo, Feng-I Chung*

# ICISA

*The Official Publication of the ICISA*

Sponsored by:

Center for e-Manufacturing and e-Commerce  
National Chung Cheng University, Taiwan

## ICISA 2011-2012 Officers

### President

Dr. Kuanchin Chen, Western Michigan University, USA

### President-Elect

Dr. Jeff Zhang, California State University at Northridge, USA

### V.P. Information Technology

Dr. She-I Chang, National Chung Cheng University, Taiwan

### Journal Co-Editors-in-Chief

Dr. Charlie Chen, Appalachian State University, USA

Dr. She-I Chang, National Chung Cheng University, Taiwan

### Treasurer

Dr. Jim Q. Chen, St. Cloud State University, USA

### Board of Directors

Dr. Shin-Yuan Hung, National Chung Cheng University, Taiwan

Dr. Albert Huang, University of Pacific, USA

Dr. Charlie Chen, Appalachian State University, USA

### International Representatives

#### Hong Kong

Dr. Benjamin Yen, University of Hong Kong

#### USA

Dr. Stephen Shih, Pennsylvania State University

#### Taiwan

Dr. Chiaming Chang, Shih Chien University

#### China

Dr. Xiaobo Wu, Zhejiang University

### Journal Advisory Board

Dr. Houn-Gee Chen, National Tsing Hua University

Dr. Eldon Li, National Cheng Chi University

Dr. James Jiang, University of Central Florida

Dr. Binshan Lin, Louisiana State University  
Dr. David Yen, Miami University

### Journal Editorial Review Board

Dr. She-I Chang, National Chung Cheng University

Dr. Stephen Shih, University of Southern Illinois

Dr. Kuanchin Chen, Western Michigan University

Dr. Michael Tam, Western Michigan University

Dr. Victor Chen, National Cheng Kung University

Dr. Zack Wong, Sonoma State University

Dr. Yiang-Liang Chen, National Central University

Dr. Kyung Hoon Yang, University of Wisconsin - LaCross

Dr. Amy Chou, Illinois State University

Dr. Jeff Zhang, California State University - Northridge

Dr. Ta-Tao Chuang, Gonzaga University

Dr. Michael Zhang, Sacred Heart University

Dr. Mark Hwang, Central Michigan University

Dr. HongJiang Xu, Butler University

Dr. June Lu, University of Houston - Victoria

**Communications of the ICISA** (CICISA, ISSN 1533-2454) is an official publication of the International Chinese Information Systems Association and is published semi-annually by the Association. All ICISA members receive one copy of the publication. Non-members may join the Association at the price of US\$20 for one year. All manuscripts submitted to the CICISA are double-blind reviewed by the normal refereeing standards.

# COMMUNICATIONS OF THE ICISA

Winter 2012  
Volume 13, Number 2

## Table of Contents

### From the President

### From the Editor

## Peer Reviewed Papers

### Learning IT Sustainability: A Balanced Scorecard Approach .....1

*Anne-Marie Guidy-Oulai, Beth Hoger, J. Michael Tarn*

### Managing Software Evolution Life Cycle: A Demand Perspective .....15

*Hsiang-Jui Kung, LeeAnn Kung*

### An Ontology Development Methodology to Integrate Existing Ontologies in an Ontology Development Process .....31

*Nelson K. Y. Leung, Sim Kim Lau, Nicole Tsang*

### Accountants' Cognition and Attitude toward the Environmental Accounting System in Taiwan Manufacturing Industry .....62

*She-I Chang, David C. Yen, I-Cheng Chang, Hsing-Jung Li, Su-Han Cheng, Lin-Wan Chen*

### Applying RFM Analysis to Library-Book Recommender Systems .....79

*Ya-Han Hu, Chia-Lun Lo, Feng-I Chung*

---

## **An Ontology Development Methodology to Integrate Existing Ontologies in an Ontology Development Process**

---

**Nelson K. Y. Leung\***

RMIT International University Vietnam, Ho Chi Minh City, Vietnam

*nelson.leung@rmit.edu.vn*

**Sim Kim Lau**

University of Wollongong, NSW, Australia

*simlau@uow.edu.au*

**Nicole Tsang**

RMIT International University Vietnam, Ho Chi Minh City, Vietnam

*nicole.tsang@rmit.edu.vn*

**ABSTRACT:** Ontology is defined as an explicit specification of a conceptualization while a conceptualization is an abstract, simplified view of the world that we wish to represent for some purpose. To build high quality ontologies, developers are required to choose and follow a suitable ontology development methodology in which containing a series of steps, activities and guidelines that are put together in an organized and systematic manner. Literatures show that building ontologies by reusing existing ontologies is more cost effective than building from scratch. However, majority of the methodologies only provide a very limited discussion about how to perform integration or ontology reused in their ontology development processes. This paper aims to develop a set of guidelines for the selection of ontology reuse methods which has not been explored in any ontology research. In addition, an ontology development methodology that incorporates ontology reuse methods and a system that can assist to perform integration semi-automatically is proposed. An application scenario has been developed to illustrate how the proposed development methodology can be adopted to create an ontology.

**Keywords:** Ontology, Merging, Integration, Ontology Development Methodology

---

\* The corresponding author.

## 1. INTRODUCTION

Ontology, a branch of philosophy, was borrowed by artificial intelligence community and defined as an explicit specification of a conceptualization while a conceptualization is an abstract, simplified view of the world that we wish to represent for some purpose (Gruber, 1993). Studer, Benjamins and Fensel (1998) further defined ontology as a machine-readable specification of a conceptualization in which the type of concepts used and the constraints on their use are explicitly defined. Ontologies also provide a shared understanding of a domain which is necessary to overcome differences in terminology from various sources (Antoniou and Harmelen, 2004). By capturing consensual knowledge accepted by large group of people, domain specific knowledge with vocabularies in terms of concepts and their interrelated describable relationships can be represented in an ontology, thus making it possible for inference engines or application programs from other intelligence system to understand its knowledge. The popularity of the WWW further magnifies the importance of ontology because of the incompetency of the HTML-based website. Computers are not able to understand and process the semantics of html-based web content because HTML is solely designed for formatting and displaying information. Semantic Web is the extension of the current one, in which web contents are represented in a structural form within ontologies by a finite list of concepts and their relationships (Berners-Lee, Hendler and Lassila, 2001). In this way, ontologies enable computer programs, software agents and search engines to understand and process the web contents.

Ontology building is sometimes considered as an art rather than a science, thus there is not a single correct ontology development methodology (Grubi and Fan, 2010; Mizoguchi, 2003; Jones, Bench-Capon and Visser, 1998). To build high quality ontologies, developers are required to choose and follow a suitable methodology in which containing a series of steps, activities and guidelines (including purpose articulation, knowledge acquisition and formalization and evaluation) that are put together in an organised and systematic manner. Although each methodology has its own characteristics and is probably more effective and efficient than the others under certain scenarios, some of them prone to agree ontology reuse is essential in their ontology development process (Fernandez-Lopez, Gomez-Perez and Juristo, 1997; Uschold and King, 1995; Noy and McGuinness, 2001; Lee, 2006; Holsapple and Joshi, 2002). Researchers pointed out that building ontology reusing existing ontologies in a development process is more cost effective than building it from scratch if reuse methods are applied appropriately (Pinto and Martins 2001). In other words, time, manpower and other resources can be saved if suitable knowledge modules are found from existing ontologies. Level of saving depends on whether the knowledge modules are required to modify before they can be reused, that is, level of saving decreases if the modules need to undergo more modification.

This paper aims to address the limitation of the existing ontology development methodologies that fail to provide detailed information on how to perform ontology reused in their ontology development process by proposing 1) a set of guidelines for the selection of ontology reuse methods, 2) an ontology development methodology that integrates ontology reuse methods, 3) a system that can assist to perform integration semi-automatically. The rest of the paper is organized as follows. Section 2 analyses a number of significant ontology development methodology. A detailed analysis of ontology integration and merging that includes guidelines for ontology reuse methods are provided in Section 3. Section 4 discusses the proposed methodology of integration-oriented ontology development and key terms evaluation system. Section 5 describes an application scenario used to illustrate the proposed ontology development methodology. Finally, conclusion is given in Section 6.

## 2. ANALYSIS OF ONTOLOGY DEVELOPMENT METHODOLOGY

In this section, we study ten significant development methodologies to investigate how ontology reused is performed in their ontology development processes. These different ontology development methodologies were established under different scenarios with various characteristics: some methodologies were derived from the experience of building a particular ontology when requirements were clearly understood at the very beginning such as Ontology-based KM Systems Methodology (Stabb, Studer, Schnurr and Sure, 2001), TOVE (Gruninger and Fox, 1995) and Enterprise Ontology (Uschold and King, 1995), some were developed using iteration approach when requirements were not clearly understood initially such as Ontology Development 101 (Noy and McGuinness, 2001), Methontology (Fernandez-Lopez et al., 1997), Collaborative Design Approach (Holsapple and Joshi, 2002) and DILIGENT (Pinto, Tempich and Stabb, 2009), some use collaborative approach to evolve ontologies according to users' usage and feedbacks (Pinto et al., 2009; Holsapple and Joshi, 2002), some were structured by combining two to three other methodologies (Uschold, 1996; Brusa, Caliusci and Chiotti, 2008) and one put their focus on using scenarios in different development activities (Lee, 2006).

Ontology development methodologies generally consist of several phases in their development process, for instance, Methontology has three phases in its process, namely pre-development, development and post-development and each phase contains a number of activities (Fernandez-Lopez et al., 1997). There are two activities that need to be executed, environmental and feasibility study, in the pre-development stage of Methontology. In the development phase, the first activity that needs to complete is to specify purpose, intended use and potential users of the ontology, followed by conceptualising the domain knowledge into meaningful knowledge-level models. After formalizing the conceptualized models into formal or semi-computable



models, the development phase can be concluded by building computable models using an ontology language. In the post-development phase, the resulting ontologies are validated and updated if needed. Besides, the authors also added a series of management (such as scheduling, control and quality assurance) and supporting activities (such as integration, merging, alignment and documentation) that can be performed throughout the development process to make sure the development process is on track.

Unlike Methontology, most of the methodologies only specify activities in the development phase. Out of the ten methodologies, only three of them covers all three phases and the remaining seven either exclude pre-development phase, post development phase or both, for example, Diligent knowledge process has merely the development phase with following activities: 1) it starts by building a core ontology using combined effort of domain experts, users, knowledge engineers and ontology engineers, 2) the users can start to use and localise the core ontology according to their own business requirements. Though they are not allowed to change the core ontology directly, they can send their change requests and local adaptations to the control board of the ontology instead, 3) the board analyses the local ontologies and change requests to decide which changes will be introduced in the next version of the core ontology, 4) the board has to revise the core ontology regularly so that the it does not diverge too far from the local ontologies, 5) the users can updated their own ontologies once a new version is released (Pinto et al., 2009). Phases, activities and other characteristics of the ten ontology development methodologies are summarised in Table 1.

Table 1 A Summary of 10 Ontology Development Methodologies

Name	Phases / Activities	Include all Phases	Ontology Reuse
Methontology (Fernandez-Lopez et al. 1997)	<ul style="list-style-type: none"> <li>• Pre-development: Environment and feasibility study</li> <li>• Development: Specification</li> <li>• Development: Conceptualization</li> <li>• Development: Formalization</li> <li>• Development: Implementation</li> <li>• Post-development: Maintenance</li> <li>• Plus additional 5 supportive activities and 3 Management activities</li> </ul>	Yes	Yes
Enterprise Ontology (Uschold and King 1995)	<ul style="list-style-type: none"> <li>• Development: Identify the purpose</li> <li>• Development: Ontology capture</li> <li>• Development: Coding</li> <li>• Development: Integrating existing ontology</li> <li>• Development: Evaluation and documentation</li> </ul>	No	Yes
TOVE (Gruninger and Fox 1995)	<ul style="list-style-type: none"> <li>• Development: Capture motivating scenarios and formalize informal competency questions</li> <li>• Development: Specify terminology of the ontology within a formal language</li> <li>• Development: Formalize the competency questions</li> <li>• Development: Specify axioms and definitions for the terms in the ontology</li> <li>• Development: Establish conditions for characterizing the completeness of the ontology</li> </ul>	No	No
Ontology Development 101 (Noy and McGuinness 2001)	<ul style="list-style-type: none"> <li>• Development: Determine the domain and scope</li> <li>• Development: Consider using existing ontologies</li> <li>• Development: Enumerate key terms</li> <li>• Development: Define classes and class hierarchy</li> <li>• Development: Define properties of classes</li> <li>• Development: Define facets of slots</li> <li>• Development: Create instances</li> </ul>	No	Yes
Scenario-based Ontology Development (Lee 2006)	<ul style="list-style-type: none"> <li>• Followed the framework of Methontology but the focus is on the role of scenarios played in each activity</li> </ul>	No	Yes
Collaborative Design Approach (Holsapple and Joshi 2002)	<ul style="list-style-type: none"> <li>• Pre-development: Preparation</li> <li>• Development: Specify initial ontology</li> <li>• Development: Identify diverse panel of participants</li> <li>• Development: Elicit critiques and comments on the ontology from panellist</li> <li>• Development: Revise the ontology to address panellists' feedback</li> </ul>	No	Yes
Unified Methodology (Uschold 1996)	<ul style="list-style-type: none"> <li>• Development: Identify the purpose</li> <li>• Development: Decide level of formality</li> <li>• Development: Identify the scope</li> <li>• Choose one approach to build ontology (Development):               <ul style="list-style-type: none"> <li>○ Build an ontology without going through above steps,</li> <li>○ Go through the above steps and begin formal encoding</li> <li>○ Go through the above steps and complete an intermediate document and an informal ontology</li> <li>○ Go through the above steps and complete a formal ontology</li> </ul> </li> </ul>	No	No
Diligent Knowledge Processes (Pinto et al. 2009)	<ul style="list-style-type: none"> <li>• Development: Domain experts, users, knowledge engineers and ontology engineers build a core ontology</li> <li>• Development: Users can use the core ontology and change according to business requirements. They are not allowed to change the core ontology but they can send their change requests to the Control Board (CB).</li> <li>• Development: The CB analyse the change requests and try to identify similarities and decide which change are going to be introduced in the next version of the core ontology.</li> <li>• Development: The CB has to revise the core ontology regularly</li> <li>• Development: Local user can update their own ontologies once the updated version is released</li> </ul>	No	No
Ontology-based KM Systems Methodology (Stabb et al. 2001)	<ul style="list-style-type: none"> <li>• Pre-development: Feasibility study</li> <li>• Development: Kick off</li> <li>• Development: Refinement</li> <li>• Development: Evaluation</li> <li>• Post-development: Maintenance</li> </ul>	Yes	Yes
Government Budgetary Ontology (Brusa et al. 2008)	<ul style="list-style-type: none"> <li>• Development: Specification</li> <li>• Development: Conceptualization</li> <li>• Development: Implementation</li> </ul>	No	No

Among the ten significant ontology development methodologies studied, majority of them have involved integrating or consider using or reuse existing ontologies as one of their development activities rather than just building an ontology from scratch (Fernandez-Lopez et al., 1997; Uschold and King, 1995; Noy and McGuinness, 2001; Lee, 2006; Holsapple and Joshi, 2002). Those methodologies recommend to examine a set of existing ontologies so that the resulting ontologies can be partially or fully constructed by reusing one or more existing ontologies in their development process. In fact, most of the ontology development methodologies only provide a shallow description about reusing existing ontologies in their development process. In most cases, authors use one section or even a few sentences to describe the definition (Holsapple and Joshi, 2002), reasons (Noy and McGuinness, 2001), difficulties (Uschold and King, 1995) and benefits of reusing existing ontologies (Stabb et al., 2001; Holsapple and Joshi, 2002). Among all, Methontology is the only methodology that goes over the process of reusing existing ontologies (Fernandez-Lopez et al., 1997). However, the downsides are the process only covers two steps (inspect meta-ontologies and search for appropriate terms from existing ontologies) and the description is too general that fails to give sufficient insight. Thus, a detailed description on how to reuse potential ontologies in an ontology development process is required.

### 3. ANALYSIS OF INTEGRATION AND MERGING

There are two possible ways to reuse ontologies in a development process, namely merging and integration. Merging creates a new ontology in one subject by unifying two or more different ontologies on that subject and it is usually hard to identify regions of the source ontologies from the merged ontologies (Noy and Musen, 1999; Pinto and Martins, 2001). Ontology integration is a process of building an ontology in one subject reusing one or more ontologies in different subjects and it is always possible to identify regions of the source ontologies from the integrated ontologies. Source ontologies may need some sort of refinements before they can be aggregated, combined and assembled together to form the resulting ontology.

Both methods can be used to build a new ontology although each of them carries some distinctive characteristics. In merging, the resulting ontology usually contains a set of more general knowledge because the knowledge of a particular subject from two or more source ontologies is generalised, extended and combined together. In integration, an ontology is selected to integrate either directly into an ontology development process if no other suitable ontology is available or into an ontology development process with other chosen ontology(ies), for example, if a help desk wants to build an ontology specifying technical problems in the area of hardware, software, network and other IT administrative issues, it just needs to identify and reuse suitable

modules from ontologies that contain the above knowledge (see Figure 1) . As relevant knowledge is available in Ontology 1 and 2, the help desk can reuse a particular module from Ontology 1 and 2 in its own ontology building process. The help desk is required to develop its own knowledge in the area of software and hardware, as there is no suitable domain knowledge available.

Although both integration and merging are capable of developing a new ontology, there is not much description about how to select merging or integration in an ontology development process. As these two methods require going through a complicated process rather than just simple revisions or improvements, whether to apply integration or merging in the development process should be considered seriously. Selecting an incorrect method may influence the development cost due to the fact that a project team may be required to go through the development process again using an alternative method. To avoid this, a set of guidelines is proposed for the selection of ontology integration and merging in the ontology development process.

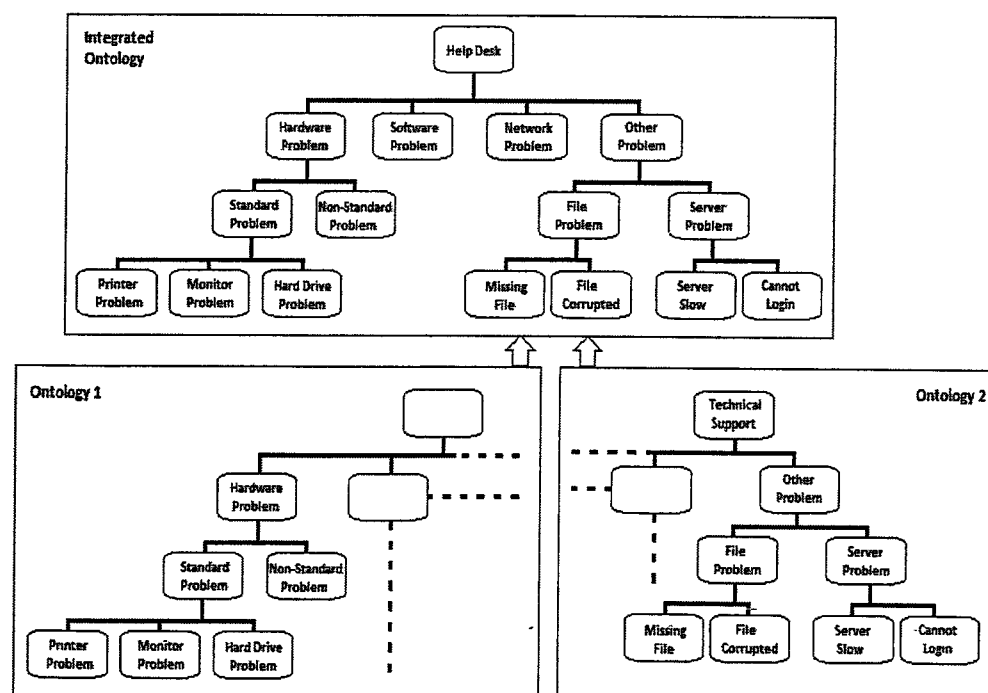


Figure 1 Example of Ontology Integration

First, merging can never take place if only one suitable ontology is available. This ontology should be used to integrate into an ontology development process instead of adopting ontology merging. Both merging and integration can be performed given that more than one ontology are available. Second, merging is more likely to be the candidate reuse method if majority of the concepts and relations from source ontologies can be used to rebuild a new ontology. Alternatively, if only one or a few portions can be reused from each source ontology, integration is more appropriate in this case. Third, merging is an ideal method to develop a top-level ontology because this type of ontology is used to describe some very general concepts (such as time, space, object, event and action) across various knowledge domains. The resulting top-level ontology is used to provide a guideline for the development of root terms that will be linked in existing ontologies. Merging is also more suitable to develop an ontology that aims to include every possible concepts and relations from multiple source ontologies. In contrast, integration should be used to develop an ontology that contains a specific type of domain knowledge by taking the required knowledge modules from source ontologies. A summary of selection guidelines is shown in Table 2.

**Table 2** A Summary of Definition and Selection Guidelines of Ontology Integration and Merging

	Merging	Integration
<b>Definition</b>	Build an ontology in one subject unifying two or more different ontologies on that subject	Build an ontology in one subject reusing one or more ontologies in different subjects
<b>Possible to Identify Source Ontology from the Resulting Ontology</b>	Not possible	Possible
<b>Selection Guidelines</b>	<ul style="list-style-type: none"> <li>• More than one source ontologies are available</li> <li>• Majority of concepts and relations from each source ontology can be reused</li> <li>• Ideal method to develop top level ontologies</li> <li>• Ideal method to develop an ontology that includes every possible concepts and relations from source ontologies</li> </ul>	<ul style="list-style-type: none"> <li>• One or more ontology is available</li> <li>• Only one or a few portions from each source ontology can be reused</li> <li>• Ideal method to develop ontologies that contain domain specific knowledge</li> </ul>

On one hand, merging is a well-developed topic in ontology research and researchers have developed quite a number of semi- or fully-automatic tools and methods in the last decade (McGuinness, Fikes, Rice and Widler, 2000; Noy and Musen, 2003; Stumme and Maedche, 2001; Maiz, Fahad, Boussaid and Bentayeb, 2010; Li, Wang and Hong, 2009). On the other hand, integration is almost an untouched ontology research area and the research has nearly stopped after the emergence of an integration methodology developed by Pinto and Martins (2001). This methodology comprises several integration activities which include feasibility study, domain

identification, candidate ontologies identification, analysis and selection, application of integration operations and resulting ontology analysis. Unfortunately, the above integration methodology has two major insufficiencies: 1) the methodology only focuses on how to find, select and integrate existing ontologies but fails to recognise integration itself is just part of an ontology development process, 2) the methodology also fails to be supported by any form of semi- or automatic tools especially in finding suitable knowledge modules from a group of candidate ontologies.

#### 4. A PROPOSED METHODOLOGY OF INTEGRATION-ORIENTED ONTOLOGY DEVELOPMENT

As mentioned in the previous sections, both the ontology development and integration methodologies have their own shortcomings. Thus, we proposed to develop the Methodology of Integration-oriented Ontology Development (MIOD) that reuses existing ontology as part of its development process. The reuse method incorporated in MIOD is integration because of its level of maturity in ontology research. The methodology not just includes a detailed description of ontology integration but also a semi-automatic tool to support the identification of suitable knowledge modules from various candidate ontologies. MIOD includes a list of activities that can help to streamline the ontology development process in a more effective and efficient manner. Some of the activities are excluded because they are redundant or non-constructive even though they are mandatory to perform in other significant development methodologies

In MIOD, the development process is divided into six phases: preparation, analysis, integration, development, implementation and evaluation, and maintenance as shown in Figure 2.

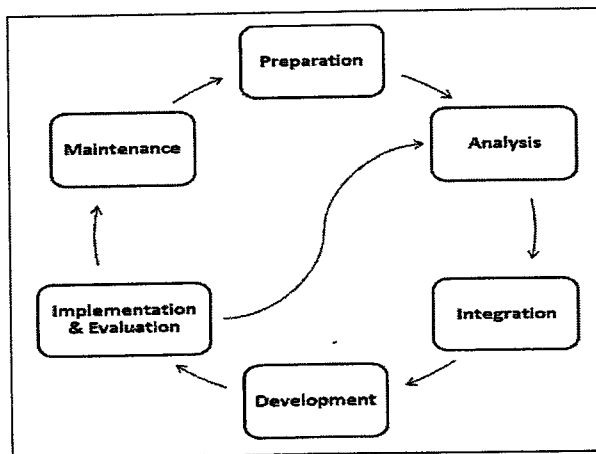
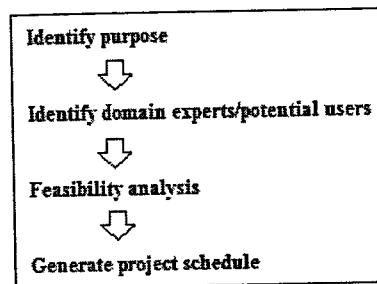


Figure 2 Six Phases of Development Process in MIOD

#### 4.1 Preparation Phase

The purpose of this phase is to identify purpose, scope, potential users and domain experts for resulting ontology, to evaluate the feasibility of the ontology development project and to generate project schedule. Four activities are required to conduct in the preparation phase as shown in Figure 3:



**Figure 3** Four Activities in Preparation Phase

The phase begins by defining the purpose of building an ontology including identification of problems that the resulting ontology will be used. The development team also needs to identify a brief scope of the ontology to indicate what must or must not be included in the resulting ontology. Other than having a project manager, the development team should also consist of system analyst, ontology engineer and knowledge engineer. A list of potential users and domain experts who possess knowledge and experience for knowledge contribution as well as ontology implementation and evaluation, and maintenance should be included too.

Feasibility analysis is an important activity for this phase. Three types of feasibility analysis are to be performed: technical, resources and financial feasibilities. The technical feasibility measures whether the proposed ontology is a practical solution among a list of other possible solutions. As ontology is always used as a component of an ontology-based application, the technical feasibility of the proposed ontology should be evaluated with the application in which the ontology will be integrated with. This feasibility can be evaluated based on performance, ease of use and efficiency of the system. Besides, the technical feasibility also evaluates time, cost and risk required to integrate the proposed ontology with the ontology-based application.

The resources feasibility evaluates if the current human and technical resources can support such a development. The financial feasibility determines if it is cost effective to continue develop the ontology by conducting an analysis of development costs and anticipated benefits. The last

activity in this phase is to generate a project schedule which includes a set of activities and their tasks, estimated duration and number of human resources needed to complete each task.

#### 4.2 Analysis Phase

The purpose of this phase is to generate and categorize key terms identified from a set of inspiration scenarios. The key terms are used to find suitable source ontologies for integration. Figure 4 shows activities that are required to be conducted in the analysis phase:

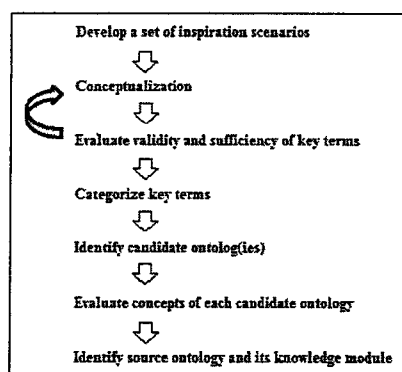


Figure 4 Seven Activities in Analysis Phase

The first activity of this phase is to develop a set of inspiration scenarios and a set of verification questions initialized by a group of domain experts. The scenarios are used to find out the type of information the ontology needs to provide. Each scenario identifies a problem that needs to be addressed by the resulting ontology. At the same time, each scenario also provides a set of possible solutions to the problem and a list of verification questions that is used to justify if the resulting ontology consists of sufficient terminologies to answer the questions. The inspiration scenarios and verification questions can be generated through discussion among members of developments, and sampling organizational and external documents. Figure 5 shows a sample template that can be used to record verification scenarios. The template includes scenario name, actors (participants) and description as well as a list of verification questions.



<b>Scenario Name</b>
<b>Actors</b>
<b>Scenario Description</b>
<b>Verification Questions</b>

**Figure 5** Sample Template of Inspiration Scenario

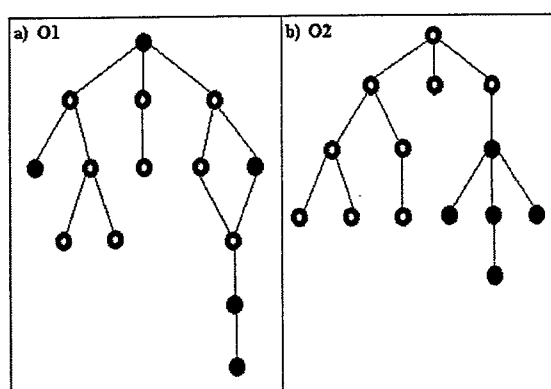
The second activity is conceptualization and the major contribution should come from potential users and domain experts. They should be encouraged to discuss in an open and friendly environment without worrying about terms may overlap, relationships among the terms and properties the terms may possess in order to capture all possible key terms related to the inspiration scenarios.

The third activity requires the development team to verify validity and sufficiency of key terms generated from the conceptualization activity. The verification can be achieved by justifying scope and level of details of the key terms to ensure enough terminologies are provided to answer all verification questions. Verification and conceptualization will iterate until all key terms are identified within the domain of interest. The forth activity categorizes key terms into subdomains. For example, key terms of IT knowledge can be categorized into software, hardware, network and other IT-related knowledge.

The fifth activity requires the development team to identify candidate ontologies in the domain of interest. Candidate ontologies can be located in ontology libraries, for instance, Protégé Ontology Library ([www.protegewiki.stanford.edu](http://www.protegewiki.stanford.edu)) and DAML Ontology Library ([www.daml.org/ontologies](http://www.daml.org/ontologies)). Alternatively they can be searched using semantic search engines such as Swoogle. Very Often, some of the partnerships or strategic alliances are willing to share their organizational knowledge as well as ontologies. At this stage, any possible ontologies should be considered as candidates and only those that have no relationship with the domain of interested are eliminated.

The sixth activity is to evaluate concepts of each candidate ontology with the categorized key terms. To achieve this, evaluations are required to conduct in all relevant candidate ontologies for each category. For each key term in a category, the development team needs to conduct a matching test to evaluate every concept from each candidate ontology in order to find semantically identical concepts. A suitable source ontology for each category can be found by comparing the number of semantically identical concepts among all candidate ontologies.

Moreover, the level of distribution among semantically identical concepts in a candidate ontology is another important consideration when choosing source ontologies. Consider O1 and O2 in Figure 6 which are two ontologies used to demonstrate how distribution of semantically identical concepts can determine the selection of a source ontology. Each node in the diagrams represents a concept where the unshaded node and shaded node are used to represent a non-identical and semantically identical concept respectively. Both O1 and O2 have fourteen concepts in which five of them are semantically identical concepts. The only difference between O1 and O2 is the level of distribution of semantically identical terms in which identical concepts in O1 are distributed more loosely than O2. In this case, rather than selecting O1 as the source ontology, O2 with the semantically identical concepts that converge closely with each other is easier to partition and modify, thus making it easier to reuse in the development process.



**Figure 6** Ontology (O1 and O2) Used to Demonstrate Distribution Level of Semantically Identical Concepts

Although the sixth activity can be done manually, such an activity can be performed semi-automatically to reduce human resources expenses. Hence, an Integration-oriented Candidate Ontology Evaluation System (ICOES) is proposed to allow the sixth activity to be carried out semi-automatically. As illustrated in Figure 7, the proposed ICOES takes a set of key terms for a particular category and relevant candidate ontologies as input. The ICOES is designed to compare each key term of a category with every possible concept in each relevant candidate ontology to check for semantically identical representation using a single or a combination of ontology matching techniques.

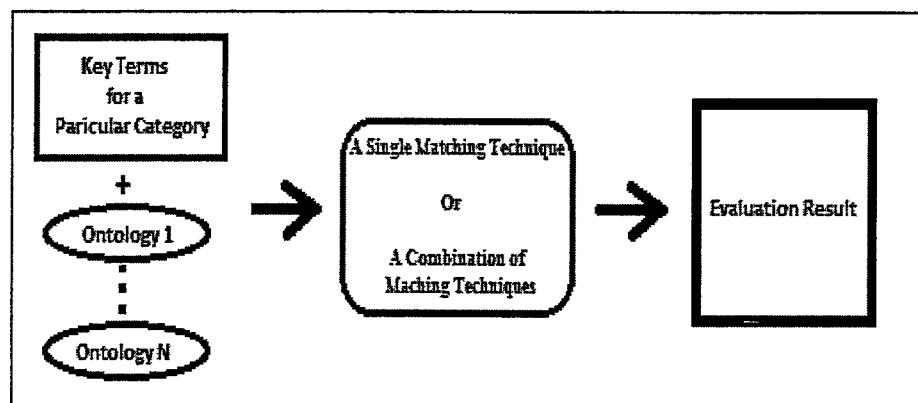


Figure 7 Proposed Integration-oriented Candidate Ontology Evaluation System (ICOES)

The 3-tier classification framework of ontology matching techniques developed by Leung, Kang, Lau and Fan (2009) provides a valuable reference point for choosing a single or combination of ontology matching techniques. Since the techniques described in the 3-tier classification framework are designed to perform matching between two ontologies, they are required to undergo modifications in order to fit in the proposed ICOES. In general, the techniques need to be modified to take key terms (identified in conceptualization activity) and concepts from a candidate ontology as input for similarity checking, instead of taking concepts extracted from two ontologies. For instance, if the ICOES uses string-based technique to find identical concepts from candidate ontologies, the technique needs to be modified in order to take key terms directly from each category. In this way, the string-based technique will be able to find semantically identical concepts from the candidate ontologies based on these key terms.

Let us demonstrate the application of the ontology matching techniques in the ICOES using string-based and linguistic source matching techniques. String-based matching technique is used in example 1 and 2 as shown in Figure 8. In example 1, the string-based matching technique

embedded in ICOES finds a concept (Network) in Ontology A that is identical to the key term "Network" from a particular list. As a result, an arbitrary matching score of one point is assigned to the pair of key term (Network) and the concept (Network).

In addition to identical representation of a key term and a concept, string-based matching technique also computes similarity if a prefix or suffix is found in an ontology that matches a particular key term. In example 2 (see Figure 8), the string-based matching technique finds a concept (Broadband Network) in Ontology B that contains the suffix of "Network". Thus, an arbitrary matching score of half a point is assigned to the pair of key term (Network) and the concept (Broadband Network).




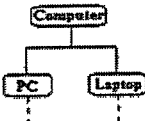
Example	Key Terms	Concepts	Top-Level Ontology	Matching Results
1	Network	 <p>Ontology A</p>		1
2	Network	 <p>Ontology B</p>		0.5
3	Computer	 <p>Ontology C</p>		0.5

Figure 8 Demonstration of Ontology Matching Techniques in ICOES

Score can be assigned based on the level of similarity. The level of similarity is higher when an identical concept is found rather than a prefix or suffix is found. Therefore, the score assigned in example 2 is lower than the score assigned in example 1 due to the fact that "Broadband Network" is not identical to but more specific than "Network"

In examples 3 (see Figure 8), the ICOES adopts linguistic resources matching technique to determine if a key term and a concept are semantically identical. Here, the technique uses domain specific thesauruses or top-level ontologies to derive a list of synonyms, hyponyms and hypernyms for every key term in a category. A score is assigned if a concept has linguistic relationship with a key term. In this example, an arbitrary score of 0.5 is assigned to the pair of

concept (Computer) and key term (PC) even though "PC" is not the same as "Computer". The score is given based on the fact that PC is a hyponym of Computer deduced from the top-level ontology.

After the matching process is completed, the proposed ICOES will generate an overall evaluation result for each pair of category and candidate ontology. The evaluation result should contain maximum, minimum and average score for each key term. Minimum and maximum scores are the minimum and maximum matching score attained for a key term against all concepts in the candidate ontology. Scores of each key term are aggregated to calculate an average matching score for a key term by dividing the aggregated matching score using the number of concepts that are semantically identical to the key term. In addition, the ICOES will also include a coverage score to indicate the average maximum similarity of all key terms.

The proposed ICOES also consists of a concept distribution counter that is used to measure the distribution level of matching concepts in the ontology. The counter views each candidate ontology as a graph with nodes (concepts) and edges (relationships) to represent the taxonomy of the ontology. For each candidate ontology, the counter has to find the shortest path (that is minimum number of nodes) for each pair of concepts that have been identified as semantically identical representations to at least one key term. To achieve this, the counter has to traverse all of the possible paths between the pair of concepts to find the total number of nodes (concepts) that are not semantically identical to any key terms in the matching result. By doing so, the counter is able to show the level of distribution for each candidate ontology by calculating the average number of unmatched nodes for each path. At the end of the matching process, the proposed ICOES also generates a distribution score to indicate the level of distribution for each pair of category and candidate ontology. Thus, the development team is able to find a suitable source ontology that can be reused in the category based on the matching scores, concept distributing scores and other evaluation criteria such as quality of documentation of the candidate ontologies and language used to implement the ontologies.

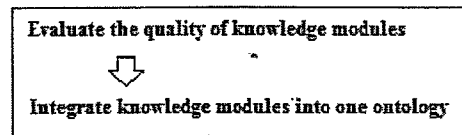
After the matching process is completed, the proposed ICOES will generate an overall evaluation result for each pair of category and candidate ontology. The evaluation result should contain maximum, minimum and average score for each key term. Minimum and maximum scores are the minimum and maximum matching score attained for a key term against all concepts in the candidate ontology. Scores of each key term are aggregated to calculate an average matching score for a key term by dividing the aggregated matching score using the number of concepts that are semantically identical to the key term. In addition, the ICOES will also include a coverage score to indicate the average maximum similarity of all key terms.

The proposed ICOES also consists of a concept distribution counter that is used to measure the distribution level of matching concepts in the ontology. The counter views each candidate ontology as a graph with nodes (concepts) and edges (relationships) to represent the taxonomy of the ontology. For each candidate ontology, the counter has to find the shortest path (that is minimum number of nodes) for each pair of concepts that have been identified as semantically identical representations to at least one key term. To achieve this, the counter has to traverse all of the possible paths between the pair of concepts to find the total number of nodes (concepts) that are not semantically identical to any key terms in the matching result. By doing so, the counter is able to show the level of distribution for each candidate ontology by calculating the average number of unmatched nodes for each path. At the end of the matching process, the proposed ICOES also generates a distribution score to indicate the level of distribution for each pair of category and candidate ontology. Thus, the development team is able to find a suitable source ontology that can be reused in the category based on the matching scores, concept distributing scores and other evaluation criteria such as quality of documentation of the candidate ontologies and language used to implement the ontologies.

The seventh activity is to choose a set of source ontologies from the group of candidate ontologies that can be reused in the development process based on the results generated from the sixth activity. The source ontologies should be the ones that exceed the predefined minimum number of semantically identical concepts (known as threshold) and consist of the highest number of identical semantically concepts in a particular category. In addition, the source ontologies should also be the ones consisting of semantically identical concepts that converge closely with each other. In other words, the source ontology should be the one that has the highest coverage and distribution scores. The seventh activity also requires to identify the appropriate knowledge modules from each source ontology. Using key terms, the development team is able to find semantically identical concepts in the source ontology and partition the most appropriate knowledge modules for each category.

### 4.3 Integration Phase

The purpose of this phase is to evaluate various knowledge modules from source ontologies and integrate them into one integrated ontology. There are two activities in the integration phase as shown in Figure 9.



**Figure 9** Two Activities in Integration Phase

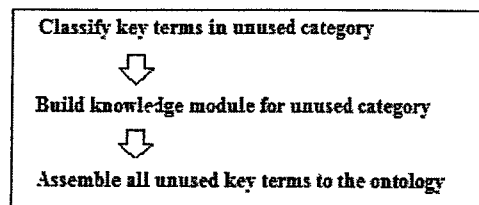
The first activity requires the development team to ensure any knowledge modules from the source ontologies are accurate, precise, simple, clear, concise, correct and complete. To achieve, each knowledge module is required to measure its completeness and way of structuring concepts using the set of categorized key terms, inspiration scenarios and verification questions. Any insufficiency may require the module to undergo some kinds of modification in names and definitions of concepts and relations, and documentation such as:

- Removal of a concept, relation, name/definition from a concept/relation or part of documentation.
- Introduction of a new concept, relation, name/definition for a concept/relation or additional content for documentation.
- Specialization of more specific concepts from a general concept.
- Generalization of a more general concept from a group of specific concepts for classification purpose.
- Standardization of a name for a concept/relation.

Modifications should only be used parsimoniously because a simple modification may cause ripple effect across an ontology. In some cases, it may be more cost effective for the development team to choose the next best ontology from a group of candidate ontologies or to build its own knowledge module. Once the quality of knowledge modules is confirmed, second activity of this phase can be performed. This includes finding a connection point for each module and building a basic ontology to connect with knowledge modules through these connection points.

#### **4.4 Development**

The development team is responsible for using key terms in unused categories to build knowledge modules as part of the integrated ontology. Unused categories are these categories identified in the analysis phase but fail to find suitable source ontologies for integration. As shown in figure 10, there are three activities included in the development phase.



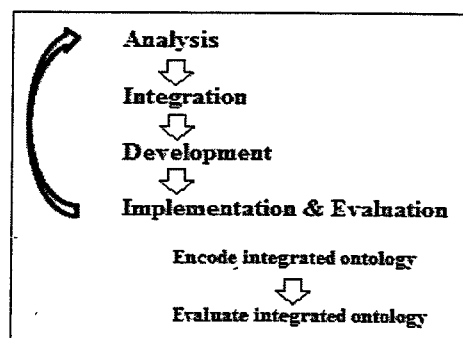
**Figure 10** Three Activities in Development Phase

In the first activity, the development team is required to use top-down or bottom-up technique to classify key terms for each unused category. Top-down technique identifies the most general key terms followed by the more specific ones, so the most general terms are placed in the first level, the more specific in the second level and so on. In contrast, bottom-up technique locates the most specific key terms followed by the general ones. After classification, each category possesses the most specific terms in the first level and the least specific in the last level.

In the second activity, knowledge modules for unused categories are built by adding relevant relations among key terms that are semantically related. The third activity requires the development team to assemble all knowledge modules for unused categories back to the integrated ontology.

#### 4.5 Implementation and Evaluation Phase

The purpose of this phase is to formalize the integrated ontology and to ensure the integrated ontology satisfies all evaluation requirements. The implementation and evaluation phase consists of two activities (see Figure 11)



**Figure 11** Two Activities in Implementation and Evaluation Phase



The first activity requires the ontology completed in the development phase be implemented using a formal representation language ranging from RDF to OWL, depending on the level of expressiveness.

The second activity requires the integrated ontology to undergo two types of evaluation. Firstly, the ontology must be able to answer all verification questions recorded in every inspiration scenario. Finally, the ontology must be evaluated in an application environment either in an actual system or a prototype and the potential users are required to provide feedbacks based on the usability of the ontology. Deficiencies or feedbacks should be reviewed and the development team may need to return to the analysis, integration, development, and implementation and evaluation phases until the ontology meets all evaluation requirements.

#### **4.6 Maintenance Phase**

An ontology needs to be updated regularly in order to maintain its validity. Maintenance phase consists of two activities:

1. Ontology Pruning.
2. Ontology Refining.

Ontology pruning is used to remove any invalid components of the ontology when components cannot be refined or are no longer needed. When the components cannot be refined, ontology refinement is needed. Ontology refinement includes extending existing components and adding new components to the ontology. Ontology refinement also includes the renewal of invalid components of the ontology including concepts, properties, relations and axioms. Level of consistency and accuracy, and content richness can be improved in the resulting ontology after ontology pruning and refinement.

Ontology may be required to modify based on changes in an ontology-based system. Ontology may also be required to change due to changes in internal and external business and political environment. Very often, the maintenance activity is performed by users who possess extensive knowledge in the domain of interest and use the ontology-based systems regularly to perform their daily task (but they may not possess high level of technical skill). When designing mechanism of ontology maintenance, it is very important to design a user interface that is appropriate for users as unfriendly user interface may hinder users from performing regular ontology maintenance.

## 5. ILLUSTRATION OF MIOD

To demonstrate the application of MIOD, let us consider the following scenario. University D, a newly established, is keen to provide excellent education and support to students but this university has limited IT budget. Although IT help desk of University D (ITD) wants to develop an ontology-based KMS that contains IT knowledge, ITD itself has insufficient expertise and human resources to support the KMS development, in particular to develop an ontology from scratch. ITD believes that the burden of ontology development can be eased by adopting MIOD as the designated development methodology because MIOD incorporates ontology integration mechanism and a semi-automatic tool (ICOES) in the development methodology. In the preparation phase, the ontology development team of ITD identifies the purpose and scope of the ontology as follows.

*Purpose: To provide a meta-knowledge layer to categorise IT help desk knowledge  
reposited in ITD's knowledge base.*

*Scope: Categories of IT admin issues, hardware and software problems and website  
errors*

Apart from the project manager, system analyst, ontology engineer and knowledge engineer, the ontology development team also includes four domain experts who possess extensive IT knowledge. The ontology development team has also conducted a thorough analysis to assess technical, resource and financial feasibilities. In the feasibility analysis, the development team finds that ontology-based KMS is the best solution among all other alternatives in terms of performance, ease of use and efficiency. The team also confirms there are sufficient human, technical and time resources as well as cost-effectiveness to conduct the ontology development project. The preparation phase is concluded by generating a project schedule.

Scenario Name: Finding solutions to web application errors
Actors: General users from various departments
Scenario Description: The scenario proposed here is a general user who is required to use a particular web application to complete his task.  The web browser fails to deliver the web application in accordance with the URL the user keyed in.
Verification Question: 1) What protocols do the web applications use? 2) What type of errors can the protocols encounter?

**Figure 12** Inspiration Scenario of "Finding Solutions to Web Application Errors"

In the analysis phase, the ontology development team is required to develop a set of inspiration scenarios that include "finding knowledge of IT admin issues", "finding hardware knowledge", "finding software knowledge" and "finding solutions to web application errors". A sample of details of inspiration scenario including scenario name, actors, scenario descriptions and verification questions are recorded in a template of inspiration scenario as shown in Figure 12.

Based on the inspiration scenarios, the development team can identify all possible key terms related to the scenarios. The key terms are verified and categorised into four categories: IT admin, hardware, software and web. Table 3 shows the set of categorised key terms identified by ITD. Moreover, the team has also identified a set of candidate ontologies from IT Solution Provider A (ITA), IT Equipment Supplier B (ITB), IT Solution Provider C (ITC) and IT Consultancy Company E (ITE).

**Table 3** Set of Categorized Keywords Identified by ITD

IT Admin	Hardware	Software	Web
PC Account Problem	Mobile Device	Microsoft Application	Web Application
PC Account Termination	Tablet Computer	Window XP	HTTP
PC Account Setup	Smart Phone	Window Vista	Client
PC Account Suspension	TV	Window 7	Server
Login Problem		DVD Player	Customize Taskbar
Email Account Problem		Projector	Customize Menu
Email Account Termination			FTP
Email Account Setup			Persistent Failure
Email Account Suspension			Transient Failure
Email Account Maintenance			Permanent Failure
Hardware Installation Guidance			
Hardware Disposal Guidance			

To evaluate candidate ontologies, the ontology development team in ITD can make use of the ICOES to determine if the candidate ontologies are suitable to be integrated as source ontologies in MIOD. As shown in Figure 13, the team has to enter a list of keywords and select a candidate ontology before evaluation can take place. After selecting the "Evaluate" function, the evaluation results are displayed.

### Evaluate ontology

Ontology will be evaluated for coverage and distribution against list of keywords

Keywords separated by comma:

PC\_Account\_Problem,  
PC\_Account\_Termination,  
PC\_Account\_Setup,

Threshold (point delimited):

0.5

☐ Simple score?

Ontology:

C:\Users\Nelson Leung

Browse...

Evaluate

**Figure 13** Sample Screen of ICOES

The ICOES is designed to generate evaluation results and the results can be divided into two types. The first result is for each key term that is entered for evaluation and it includes min score, max score, average score, number of matches and score variance (see Figure 14). Min and max scores are the minimum and maximum similarity score attained for a key term against all concepts in the candidate ontology and the value of the scores can be influenced by specifying a threshold. Average score is calculated by dividing the total similarity score of a key term by the number of matching concepts for which similarity score is greater than zero whereas score variance is the variance of similarity measure for the key terms.

The second result is the overall results for all key terms against the candidate ontology that includes coverage and distribution (see Figure 14). In the output, coverage is the average maximum similarity (max score) of all key terms and distribution is the average number of unmatched concepts between a pair of matched concepts using the shortest path. The team can choose to display all of the above results or to include only the coverage, distribution and max score for each key term. A summary of evaluation results conducted by ITD is illustrated in Table 4. Based on the results in this instance, it is appropriate to select ITA's ontology, ITE's ontology, ITB's ontology and ITC's ontology for IT admin, software, hardware and website respectively as source ontologies as their coverage scores are the highest in the categories. In some cases, if the coverage scores are very close, then the distribution scores will be used to determine the most appropriate source ontology.

Ontology file	ontology1.owl				
Threshold	0.5				
Coverage	0.6764				
Distribution	3.0				
Keyword	Min score	Max score	Average score	# matches	Score variance
PC_Account_Problem	0.5	0.6471	0.5387	12	0.0014
PC_Account_Termination	0.5789	0.76	0.6695	2	0.0082
PC_Account_Setup	0.5143	0.6875	0.6094	4	0.006
PC_Account_Suspension	0.5946	0.75	0.6723	2	0.006
Login_Problem	0.5	0.6957	0.5754	25	0.0029
Email_Account_Problem	0.5	0.7	0.572	3	0.0082
Email_Account_Termination	0.6364	0.717	0.6767	2	0.0016
Email_Account_Setup	0.5	0.7368	0.5963	4	0.0093
Email_Account_Suspension	0.6512	0.7059	0.6785	2	7.0E-4
Email_Account_Maintenance	0.6364	0.717	0.6767	2	0.0016
Hardware_Installation_Guidance	0.7123	1.0	0.8374	6	0.0085
Hardware_Disposal_Guidance	0.0	0.0	0.0	0	0.0

Figure 14 Sample Screen of Evaluation Results of IT Admin Key Terms and ITA's Ontology

Table 4 Summary of Ontology Evaluation for ITD

Ontology	IT Admin	Hardware	Software	Website
ITA	Coverage: 0.674 Distribution: 3.0	Coverage: 0.132 Distribution: 0	Coverage: 0 Distribution: 0	Coverage: 0.309 Distribution: 1.7
ITB	Coverage: 0 Distribution: 0	Coverage: 0 Distribution: 0	Coverage: 0.917 Distribution: 0.8	Coverage: 0 Distribution: 0
ITC	Coverage: 0 Distribution: 0	Coverage: 0.762 Distribution: 1.3	Coverage: 0 Distribution: 0	Coverage: 0 Distribution: 0
ITE	Coverage: 0 Distribution: 0	Coverage: 0 Distribution: 0	Coverage: 0.095 Distribution: 0	Coverage: 0.677 Distribution: 0.786

Based on the evaluation results generated by the ICOES, the development team has selected all four candidate ontologies as the source ontologies. The development team has also identified the appropriate knowledge module from each source ontology. Figure 15a, 15b, 15c and 15d show the knowledge module partition from ITA, ITB, ITC and ITE respectively.

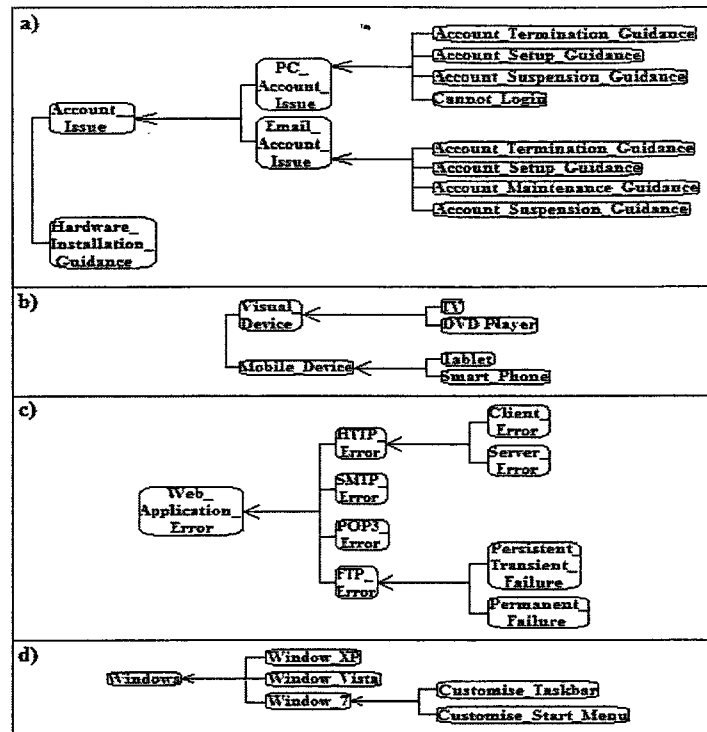


Figure 15 Knowledge Module of Source Ontology from a) ITA, b) ITB, c) ITC and d) ITE

In the integration phase, the team is required to evaluate the quality of the knowledge modules from the source ontologies using the set of categorised key terms, inspiration scenarios and verification questions. The development team has renamed some of the concepts to match with the key terms identified in the analysis phase. Additional concepts are also added to ensure completeness of the knowledge modules. This phase is concluded by integrating the knowledge modules into an integrated ontology as shown in Figure 16.

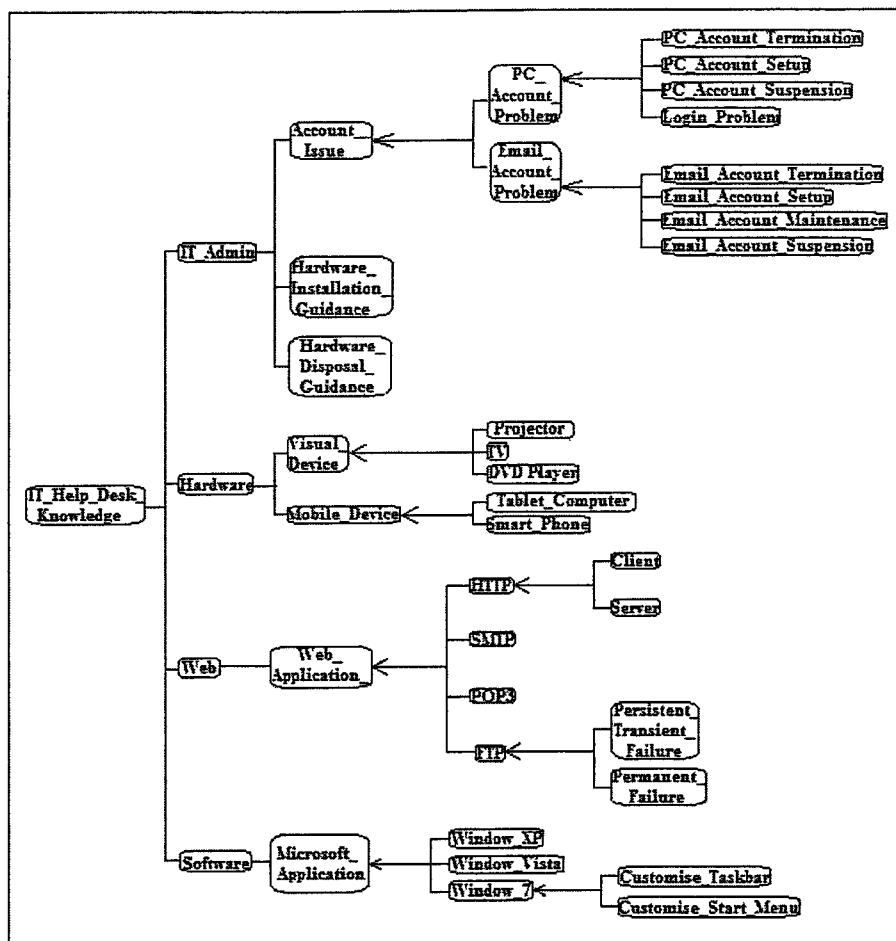


Figure 16 Sample of the Integrated Ontology of ITD

As there is no unused key terms left, it is not necessary to conduct the development phase. In the implementation phase, the development team encodes the integrated ontology into OWL. The formalized ontology is then evaluated in the application environment by a group of general users. When the users are satisfied with the performance of the KMS and the ontology, the ontology will be signed off and deployed. From this point onward, the ontology enters the maintenance phase and regular update is required to maintain its validity.



## 6. CONCLUSION

As ontologies are the building block of the Semantic Web, the popularity of the semantic web has further magnified the importance of ontology. Organizations and individuals have built ontologies of different domains that are sharable. Thus it is not impractical to build an ontology by reusing existing ontologies that are available publicly. This paper has proposed and developed guidelines on how to select ontology merging or integration in a development process.

Literatures show that building ontologies by reusing existing ontologies is more cost effective than building from scratch. Our research findings also show that the majority of methodologies have involved integrating or reusing ontology as one of their development activities. However, these methodologies only provide a very limited discussion about how to perform integration or ontology reused in their ontology development processes.

Pinto and Martins (2001) have developed an integration methodology consisting of several activities to perform ontology integration. But the methodology only focuses on how to find, select and integrate existing ontologies but fails to recognize integration itself is part of an ontology development process. The integration methodology also fails to keep up with current environment as well as incorporate any forms of technology in its building process.

To address the above limitations, an ontology development methodology called MIOD is proposed that includes preparation, analysis, integration, development, implementation and evaluation, and maintenance phases in its development process. MIOD provides a detailed description on how to incorporate and perform integration within an ontology development process including elicitation of key terms, identification of source ontologies and their knowledge modules, and application of ontology integration.

Following the MIOD, an ontology development team is required to identify purpose, scope, potential users and domain experts for resulting ontology, to evaluate feasibility of the ontology development project and to generate project schedule. In the analysis phase, the team has to generate and categorize key terms identified from a set of inspiration scenarios. The key terms are then used to find suitable source ontologies for integration.

In integration phase, the team is required to evaluate various knowledge modules from source ontologies and integrate into one ontology. In the development phase, the team has to use key terms in unused categories to build knowledge modules. These knowledge modules are assembled to the integrated ontology. In implementation and evaluation phase, the team is responsible for formalizing the integrated ontology and ensuring that the integrated ontology

satisfies all evaluation requirements. In the maintenance phase, the team has to update the ontology regularly in order to maintain its validity.

In addition, a semi-automatic tool called ICOES is also proposed that can be used to find suitable source ontologies from a group of candidate ontologies using a single or a combination of matching techniques as well as concept distribution counter. The concept distribution counter is designed to measure the distribution level of matching concepts in the ontology. Based on the matching results generated by the ICOES, the team can select the most appropriate candidate ontologies as source ontologies.

To illustrate how MIOD and ICOES can be adopted to create an ontology, an application scenario has been developed. In the scenario, ITD is able to build ontology reusing existing ontologies by the adoption of MIOD and ICOES. In fact, MIOD can be used to create ontologies in any circumstances by adopting a single or a combination of ontology matching techniques in ICOES. It is more appropriate to use a single ontology matching technique if ontologies to be integrated are simple and lightweight. However, a set of multiple matching techniques, aggregation algorithm and cut-off point must be used for integrating complex and heavyweight ontologies. In conclusion, MIOD provides tools and mechanisms that enables organizations to select appropriate sources ontology(ies) and integrate modules of the source ontology(ies) in an ontology development process.

## REFERENCES

1. Antoniou, G. & Harmelen, F. (2004). *A Semantic Web Primer*. London: The MIT Press.
2. Berners-Lee, T., Hendler, J. & Lassila, O. (2001, May). The Semantic Web. *Scientific American* [Online]. Available: <http://www.scientificamerican.com/article.cfm?id=the-semantic-web>.
3. Brusa, G., Caliusco, M. & Chiotti, O. (2008). Towards Ontological Engineering: a Process for Building Domain Ontology from Scratch in Public Administration. *Journal of Knowledge Engineering*. 2(5), 484-503.
4. Fernandez-Lopez, M., Gomez-Perez, A. & Juristo, N. (1997). Methontology: From Ontological Art towards Ontological Science. *Spring Symposium on Ontological Engineering of AAAI* (pp. 37-40). California, USA.
5. Gruber, T. (1993). Toward Principles for the Design of Ontologies Used for Knowledge Sharing. *Formal Ontology in Conceptual Analysis and Knowledge Representation*. Kluwer Academic Representation.

6. Grubic, T. & Fan I. (2010). Supply Chain Ontology: Review, Analysis and Synthesis. *Computers in Industry*. 61(8), 776-786.
7. Gruninger, M. & Fox, M. (1995). The Role of Competency Questions in Enterprise Engineering. *Proceedings of the IFIP WG5.7 Workshop on Benchmarking - Theory and Practice*. Trondheim, Norway.
8. Holsapple, C. & Joshi, K. (2002). A Collaborative Approach to Ontology Design. *Communication of ACM*. 45(2), 42-47.
9. Jones, D., Bench-Capon, T. & Visser, P. (1998). Methodologies for Ontology Development. *Proceedings of Information Technology and Knowledge Systems (15th IFIP World Computer Congress)*.
10. Lee, J. (2006). The Roles of Scenario Use in Ontology Development. *Knowledge and Process Management*. 13(4), 270-284.
11. Leung, N. K. Y., Kang, S. H., Lau, S. K. & Fan, J. (2009). Ontology Matching Techniques: a 3-Tier Classification Framework. *International Journal of the Computer, the Internet and Management*. 17(SP1), 22.1-22.7.
12. Li, S., Fan, H., Wang, Y. & Hong, L. (2009). A Model and Approach for Heterogeneous Ontology Automatic Merging. *Proceedings of International Joint Conference on Computational Sciences and Optimization* (pp. 214-217).
13. Maiz, N., Fahad, M., Boussaid, O. & Bentayeb, F. (2010). Automatic Ontology Merging by Hierarchical Clustering and Inference Mechanisms. *Proceeding of I-KNOW* (pp. 81-93). Graz, Austria.
14. McGuinness, D., Fikes, R., Rice, J. & Widler, S. (2000). An Environment for Merging and Testing Large Ontologies. *KR2000: Principles of Knowledge Representation and Reasoning* (pp. 483-493). San Francisco, USA.
15. Mizoguchi, R. (2003). Tutorial on Ontological Engineering: Introduction to Ontological Engineering (Part 1). *New Generation Computing*. 21(4), 362-383.
16. Noy, N. & McGuinness, D. (2001). *Ontology Development 101: A Guide to Creating your First Ontology*. California: Stanford Knowledge Systems Laboratory.
17. Noy, N. & Musen, M. (1999). Smart: Automated Support for Ontology Merging and Alignment. *Proceedings of 12<sup>th</sup> Workshop on Knowledge Acquisition, Modeling and Management*. Albert, Canada.
18. Noy, N. & Musen, M. (2003). The PROMPT Suite: Interactive Tools for Ontology Merging and Mapping. *International Journal of Human-Computer Studies*. 59(6), 983-1024.
19. Pinto, H. & Martins, J. (2001). A Methodology for Ontology Integration. *Proceedings of the 1st International Conference on Knowledge Capture* (pp. 131-138). British Columbia, Canada.

20. Pinto, H., Tempich, C. & Staab, S. (2009). Ontology Engineering and Evolution in a Distributed World Using DILIGENT. *Handbooks on Ontologies* (pp. 153-175).
21. Stabb, S., Studer, R., Schnurr, H. & Sure, Y. (2001). Knowledge Processes and Ontologies. *IEEE Intelligent Systems*. 16(1), 26-34.
22. Studer, R., Benjamins, V. & Fensel, D. (1998). Knowledge Engineering: Principles and Methods. *Data and Knowledge Engineering*. 25, 161-197.
23. Stumme, G. & Maedche, A. (2001). FCA-MERGE: Bottom-Up Merging of Ontologies. *Proceedings of the 7<sup>th</sup> International Joint Conference on Artificial Intelligence* (pp. 225-230). Washington, USA.
24. Uschold, M. (1996). Building Ontologies: Towards a Unified Methodology. *Proceedings of Expert Systems (Annual Conference of the British Computer Society Specialist Group on Expert Systems)*. Cambridge, UK.
25. Uschold, M. & King, M. (1995). Towards a Methodology for Building Ontologies. *Proceedings of Workshop on Basic Ontological Issues in Knowledge Sharing (International Joint Conferences on Artificial Intelligence)*. Montreal, Canada.