PAPER

A Collaborative Framework Supporting Ontology Development Based on Agile and Scrum Model

Akkharawoot TAKHOM[†,††a)], **Sasiporn USANAVASIN**[††b)], **Thepchai SUPNITHI**[†c)], ***Members***, ***and*** **Prachya BOONKWAN**[†], ***Nonmember***

SUMMARY Ontology describes concepts and relations in a specific domain-knowledge that are important for knowledge representation and knowledge sharing. In the past few years, several tools have been introduced for ontology modeling and editing. To design and develop an ontology is one of the challenge tasks and its challenges are quite similar to software development as it requires many collaborative activities from many stakeholders (e.g. domain experts, knowledge engineers, application users, etc.) through the development cycle. Most of the existing tools do not provide collaborative feature to support stakeholders to collaborate work more effectively. In addition, there are lacking of standard process adoption for ontology development task. Thus, in this work, we incorporated ontology development process into Scrum process as used for process standard in software engineering. Based on Scrum, we can perform standard agile development of ontology that can reduce the development cycle as well as it can be responding to any changes better and faster. To support this idea, we proposed a Scrum Ontology Development Framework, which is an online collaborative framework for agile ontology design and development. Each ontology development process based on Scrum model will be supported by different services in our framework, aiming to promote collaborative activities among different roles of stakeholders. In addition to services such as ontology visualized modeling and editing, we also provide three more important features such as 1) concept/relation misunderstanding diagnosis, 2) cross-domain concept detection and 3) concept classification. All these features allow stakeholders to share their understanding and collaboratively discuss to improve quality of domain ontologies through a community consensus.

key words: *collaborative activities, ontology development, standard, agile, scrum, process engineering*

1. Introduction

Ontology [1] is the essential Semantic Web technology to describe concepts and relations in order to express knowledge, especially in a specific domain. The benefits of ontologies [2] play prominent roles in knowledge representation, sharing, and also simplifying ontology-based application development.

Several research studies [3] have been purposed frameworks and tools for ontology modeling and editing. Necessary essential components [4] have been designed to support an ontology development environment. In trends of developing the environment with supporting components, one of the challenge tasks is to design and develop an ontology regarding theories of ontology modeling. Through a cycle of ontology development, this challenge is quite similar to software development as it requires collaborative activities from various stakeholders, such as domain experts, knowledge engineers, application users, etc.

Most of the existing tools did not provide collaborative features to support stakeholders in their collaboration more effectively. There is some lack of standard process adoption for ontology development. Thus, in this work, we incorporated processes of ontology development into Agile model, as used of process standard in software engineering. Based on Scrum model, we can perform to organize working stages of collaborative activities that can reduce the development cycle as well as it can be responding to any changes better and faster.

To support this idea, we aim to enhance a collaborative ontology development framework based on Agile [5] and Scrum [6] models, which systematize processes of collaborative activities and organize working stages of stakeholders. Each process of ontology development based on Scrum model will be supported by different services in our framework, aiming to promote collaborative activities among different roles of stakeholders.

However, ontology engineering is a challenging task due to multiple expert stakeholders involved in the process. We deem the software development processes as a means to accomplish this task. In software engineering, collaborative efforts coalesce in developing a software system from cradle to grave. This process is comparable to ontology engineering, in which multidisciplinary fields of knowledge have to be integrated.

In this paper, we chose the development of a Buddha image ontology as an exemplar case of ontology engineering. Analogically speaking, the body of knowledge is equivalent to a software system. Archaeologists, ancient architects, theologists, and experts on folklores and ethnic cultures collaborate in a Waterfall Model-style process of knowledge sharing, development, and curation. The outcome of this process is an ontology of Buddha image characteristics.

In addition, we support services of ontology visualized

Manuscript received February 28, 2020.
Manuscript revised July 8, 2020.
Manuscript publicized September 4, 2020.

[†]The authors are with National Electronics and Computer Technology Center, 112 Thailand Science Park, Phahonyothin Rd., Khlong Luang, Pathum Thani, 12120 Thailand.

[††]The author is with Sirindhorn International Institute of Technology, 131 Moo 5, Tiwanon Rd., Bangkadi, Muang, Pathum Thani 12000 Thailand.

a) E-mail: akkharawoot.tak@ncr.nstda.or.th (Corresponding author)
b) E-mail: sasiporn.us@siit.tu.ac.th (Corresponding author)
c) E-mail: thepchai@nectec.or.th (Corresponding author)

DOI: 10.1587/transinf.2020EDP7041

Copyright © 2020 The Institute of Electronics, Information and Communication Engineers

modeling and editing in three features: 1) concept/relation misunderstanding diagnosis, 2) cross-domain concept detection, and 3) concept classification. All these features allowed stakeholders to share their understanding and collaboratively discuss for improving the qualification of domain ontologies through a community consensus.

The main contributions of our work are as follows: (1) the proposed processes based on Agile and Scrum models to describe processes of ontology development. (2) the proposed framework provided supporting features to support collaborative ontology development, and to facilitate stakeholder's discussion.

The rest of the paper is organized as follows: Sect. 2 discussed important related works and Sect. 3 introduces a proposed architecture of ontology development framework based on Agile and Scrum model. This section explains how the framework can be useful in processes of ontology development and support in collaborative activities in ontology design. In Sect. 4, a collaborative use-case scenario demonstrates the development of domain ontology in a cultural domain. Section 5 explains a measurement of related works and our framework and discuss important issues. Lastly, Sect. 6 concludes the main contributions and gives an outlook on further work.

2. Background and Related Works

This section discusses two essential engineering concepts (software and ontology engineering) and related works.

2.1 Ontology Development and Software Engineering

Ontology [1] is the essential technology of the Semantic Web associated with comprehensive terminologies. It enables the application of knowledge-based constraints during the process of extraction and interpretation. In case of representing knowledge in a specific area, *domain ontologies* [2] play a prominent role in achieving interoperability across heterogeneous domain and systems.

Developing useful domain ontologies requires in-depth knowledge of stockholders, as a source of knowledge. ***Ontology engineering*** (***OE***) [7] is a research field concerning ontology development processes, methods, and methodologies in the management of ontologies life cycle. The OE is helpful to solve the interoperability problems brought by semantic obstacles related to the definitions of business terms and software development.

OE environments [3] have been a trend to provide essential tools with necessary components [4], such as offering many desirable services of ontology development to create classes, properties, or visualization. Many researchers have designed and implemented the OE environments for significant attention accorded to Semantic Web editor. For example, OntoEdit [8] is an ontology development environment that integrates a collaboration of various stakeholder aspects and allows them to identify requirements, refinement, and maintenance.

Although the OE environments provide supportive services, activities of various stakeholders need to be organized their collaboration. Thus, an approach of **software engineering** (**SE**), called *Agile model* [5], can organize the stakeholders to underly their requirements and solutions evolve through the collaborative effort of a self-organizing and cross-functional teams and domain-ontology users.

Several methodologies of knowledge engineering based on Agile model have been proposed as development of domain ontologies. First, H. Knublauch [9] purposed *XP.K* (*eXtreme Programming of Knowledge-based systems*). The methodology was used for knowledge modeling for supporting a close collaboration between domain experts and engineers. Second, S. Auer and H. Herre [10] purposed *RapidOWL* bringing a stable state of the knowledge base through small incremental changes from a multiplicity of contributors. Third, D. Parsons [11] purposed *AOSD* (*Aspect-Oriented Software Development*) that used ontology mapping. This work proposed an analytical framework to understand how AOSD with Agile methods might usefully be integrated into the SE approach. Fourth, S. Peroni [12] introduced *SAMOD* (*Simplified Agile Methodology for Ontology Development*) which is a methodology to develop ontologies in small steps of an iterative workflow. SAMOD focused on creating well-developed and documented models starting from exemplar domain descriptions. Recently, A. S. Abdelghany el al. [13] purposed *AMOD* (*an Agile Methodology for Ontology Development*) which is a methodology of ontology development from principles and activities of software development. This work also evaluated existing OE methodologies.

In this paper, we exploit *Scrum model* [6] that is a crucial methodology of *Agile model* [5] to manage and maintain working systems of domain ontologies, as a production prototype for relevant stakeholders.

2.2 Ontology Application Management

Many domain ontologies have been established according to domain experts from various fields. Stakeholders need a high learning curve and efforts demanded in building semantic web and ontology-based applications in case they need to employ the domain ontologies. Buranarach M. et al. [14] provided an *ontology application management* (*OAM*) framework to support stakeholders in simplifying the development of ontology-based applications. The framework provided a graphical ontology editor, called Hozo [15], in knowledge-bases visualization. The editor allows stakeholders to represent knowledge in forms of domain concepts and concept properties.

2.3 Collaborative Ontology Development

Although many related works have been provided ontology development frameworks, a collaborative approach also required as a medium function to work with system users. The following two major frameworks provided functions for collaborative development of ontologies. First, WebProtégé [16] is a lightweight ontology editor that a Web tool for knowledge acquisition. This work enabled users to create custom knowledge-acquisition forms tailored for domain experts. Second, Hozo [17] is an ontology editor that provided a collaborative function for resolving the issue based on management of dependencies between ontology modules. This work supports cooperative ontology construction based on modularization, and focus on the phase of construction and maintenance and discuss a framework, such as constructing a single ontology by many developers.

Even though the major frameworks provided collaborative functions supporting knowledge-acquisition, construction, maintenance, and discussion, some of the essential functions also require to enhance stakeholder's collaboration. For this reason, our approach focuses on communication and cross-disciplinary detection that is the exploitation of Agile and Scrum models. The framework in this paper contributes a guideline supporting the ontology development process based on standard guidelines as Agile model and organizing collaborative activities based on Scrum model. Our collaborative framework allows stakeholders to design cross-domain concepts/relations and to discuss other stakeholders in multidisciplinary knowledge.

For willing stakeholders to contribute and share their knowledge, the OAM framework was enhanced in collaborative knowledge sharing and knowledge co-creation by different stakeholders (e.g., domain experts and knowledge engineers), named a *community-driven ontology-based application management* (*CD-OAM*) framework [18]. The improved framework facilitates stakeholders in knowledge construction of multiple domains. This work presented a collaborative approach allowing stakeholders to understand cross-domain knowledge and to create/modify/extend/reuse the multidisciplinary ontology [19].

The methodology based on Agile and Scrum models has remaining challenges [20]. Working with a collaborative framework has the remaining drawbacks in the following challenges: (1) organizing processes of ontology development and (2) managing working stages for collaborative activities of stakeholders.

To tackle the challenges, in this paper, we intend to enhance the CD-OAM framework [18] by employing key benefits of Agile and Scrum models.

First, a development approach based on **Agile model** [5] is to enhance collaborative activities of ontology development. The Agile prioritizes communication of relevant stakeholders and improve processes of ontology development. The key benefits of Agile designate working directions in the following: (1) focusing more on stakeholder interaction, (2) qualifying domain ontologies, and (3) responding suggestions and comments of stakeholders.

Second, Scrum is an agile framework for the definition, execution, and delivery of the project outcome. To systematize working stages of stakeholders, the essential advantages of **Scrum model** [6] are exploited as follows: (1) to identify working directions and ontology capabilities, (2) to provide value-added to domain ontologies, (3) to deliver relevant domain ontologies to work with goal and application of stakeholders, and (4) to sustain employing of domain ontology life cycle.

3. Ontology Development Based on Agile and Scrum Models

3.1 An Overview of System Architecture

The CD-OAM framework [18] was designed to incorporate all the modules with necessary services of developing domain ontologies and offering communicability to stakeholders who want to share their understanding with others.

To improve the framework, the system was designed and implemented based on the Laravel framework [21], [22]. Development of the framework followed the model-view-controller (MVC) [23] architectural pattern and worked on a client-server model. Thus, system users can manipulate domain ontologies on a client-side, and user interactions are done on a client-side.

As illustrated in Fig. 1, the CD-OAM framework is simplified in a multi-tier architecture that represents a client-server architecture of the necessary services. Five different tiers of the architecture contain supporting modules represented in three different colors. First, the green modules belonging to Agile model support the systematization of processes for ontology development. Next, the yellow modules belonging to Scrum model support the management of working stages for collaborative activities of stakeholders. Last, the gray modules facilitate the management of database and knowledge-base. Each tier is elaborated those supporting modules as follows.

- At *the user tier*, the first tier of the framework allows stakeholders to assess web applications.

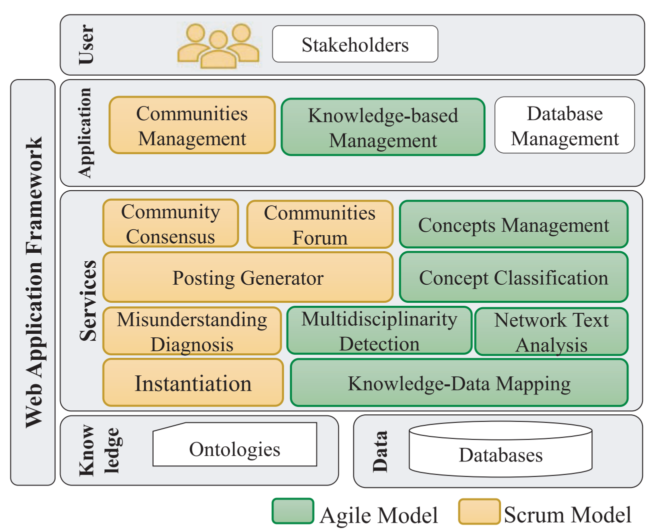


Fig. 1 The CD-OAM framework [18] based on Agile and Scrum models is represented in a multi-tier architecture. Note that green modules belong to Agile model, and yellow modules belong to Scrum model.

- At *the application tier*, the second tier provides three modules for user interaction. For Agile model, the module knowledge-based management allows users to view, upload, and create domain ontologies. For Scrum model, the module of community management allows users to communicate through a discussion forum. The module of database management supports to define, manipulate, retrieve and manage data in a database.
- At *the service tier*, the third tier contains many service modules that can be invoked over the network to processes web-based applications. Based on Agile model, the green modules support processes of collaborative ontology development including concept management, concept classification, multidisciplinary detection, network text analysis, and knowledge-data mapping. Based on Scum model, the yellow modules support the management of working stages for stakeholders including community consensus, community forum, post generator misunderstanding diagnosis, and instantiation.
- At *the knowledge tier*, this tier stores ontologies, as resources of knowledge that are created and uploaded by system users.
- At *the data tier*, this tier stores data from the module of community management, such as user profile information, comments, and suggestions from domain communities. For example, instance entities are integrated into a big linked domain knowledge network, as a result of multidisciplinary detection.

3.2 Ontology Development Based on Scrum Model

To process ontology development, Noy and McGuinness [24] provided the instruction to analyze characteristics of domain knowledge and create domain ontologies. The CD-OAM framework was designed by employing the instruction, as mentioned in Sect. 2.3.

Figure 2 illustrates a workflow of the CD-OAM framework enumerating an interrelation of SE and OE. At the top part of Fig. 2, the green areas cover processes of ontology development based on Agile model. Each green rectangle represents the *Process* that is organized into three phases: pre-game, development, and post-game. At the bottom part of Fig. 2, the yellow areas cover the working stages of collaborative activities following Scrum model. Each yellow rectangle represents a working stage that interacts with five types of stakeholders (blue rectangles) consisting of an ontology owner, domain experts, knowledge engineers, members of domain communities, and application users. Each developing process of the workflow is elaborated in the following explanations.

The first developing process (*Process 1*) is to define a domain of knowledge. As the working stage of a *Scrum owner*, an ontology creator has to define objectives of knowledge construction. The creator needs the define all possible features of *ontology backlog*, as inputs of Scrum

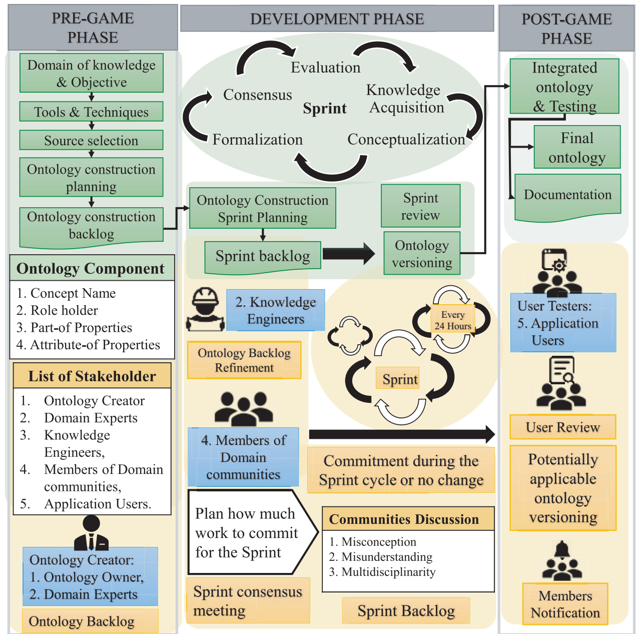


Fig. 2 A workflow of the CD-OAM framework [18] is organized into three phases: pre-game, development, and post-game. Three key components of the framework are: (1) based on Agile model, the processes of ontology development (green rectangles) are covered in green areas, (2) based on Scrum model, the working stages of collaborative activities (yellow rectangles) are covered in the green areas, and (3) five types of stakeholders (blue rectangles) exist in each working stage of the Scrum model.

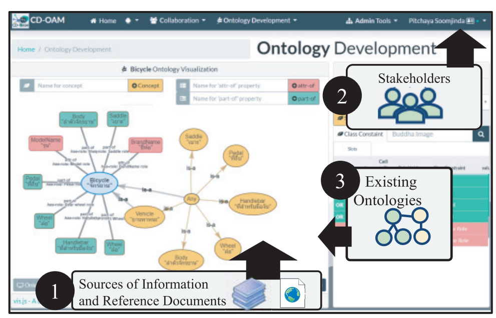


Fig. 3 Excerpts of an ontology development module consist of three sources of knowledge: (1) sources of information and reference document, (2) stakeholders, and (3) existing domain ontologies.

workflow.

Next, the second developing process (*Process 2*) is to select related tools and techniques for achieving the development of software or ontologies. As illustrated in Fig. 3, a module of knowledge-based management provides a web-based application allowing stakeholders to develop domain ontologies. They can use an ontology editor and visualization to develop ontologies based on a role-modeling theory that needs to define four components: concept names, role holders, part-of properties, and attribute-of properties.

After that, the third developing process (*Process 3*) is to select sources of knowledge. Figure 3 illustrates three essential sources of knowledge, and each source is detailed as follows. The first source (1) is information from the in-

Table 1 A pattern of user story consists of role, goal, benefit, and acceptance criteria. Note that this pattern presents the requirements of "Buddha Image" knowledge from a domain expert.

User Story of an Ontology Creator	
Role:	As a domain expert,
Goal:	I share my knowledge about "Budda Image" and make an understanding for stakeholders from other field experts in a period of creation.
Benefit:	So that we can analyze components of "Budda Image" to find a connection of a period of creation.
Acceptance Criteria:	Domain experts in a cultural field, especially "Budda Image" need to develop a domain ontology based on their understanding and knowledge.

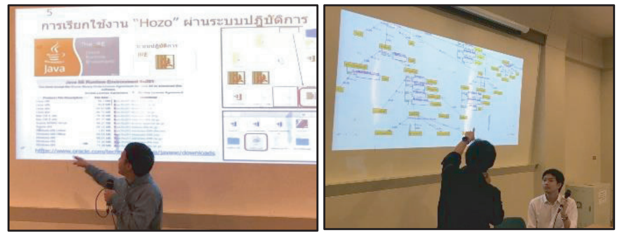


Fig. 4 The workshop of Semantic Web technology for developing domain ontology at Chiang Mai international convention and exhibition center on February 2, 2019.

ternet or reference documents, e.g., e-books or links to the website. The second source (2) is stakeholders who create domain ontologies and develop ontology-based applications [18]. They can comment and suggest other domain experts. The third source (3) is to re-use existing domain ontologies. For example, to represent multidisciplinary knowledge [25], stakeholders can create "Bicycle" ontology, and use it to develop an ontology-based application for recommending a good bicycle for tourism.

The fourth developing process (*Process 4*) is to plan for ontology construction. The creator needs to design a domain ontology. This process refers to *Sprint* that is an iterative process for the life cycle of software/ontology development.

The fifth developing process (*Process 5*) is to prepare ontology construction backlog. In each Sprint planning meeting (*Process 6*), stakeholders provide a user story of ontology construction consisting of role, goal, benefit, and acceptance criteria. Table 1 shows an example of a user story.

The seventh developing process is Sprint review (*Process 7*). In each meeting, *members of domain communities* need to make a commitment. This reviewing process has a cycle of *Sprint* for ontology development consisting of *knowledge acquisition*, *conceptualization*, *formalization*, *consensus*, and *evaluation*. The output of this developing process provides *Sprint Backlog* (*Process 8*) and keep *ontology versioning* (*Process 9*).

The last developing process (*Process 10*) is to integrate the *final ontology* and test by using the OAM framework to develop a simple ontology-based application. In this process, the relevant stakeholders have to make *documentation*.

4. A Collaborative Use-Case Scenario

The collaborative use-case scenario explains ontology development for a cultural domain as "Buddha Image". The domain refers to three-dimensional images of the lord Buddha made from stone, wood, clay, or metal. In Thailand, a sacred statue of the Lord Buddha had been found since the 3rd–4th century created unique styles from the 7th century until now. Each style of statues reflects craftsman uniqueness, artistic development, prosperity, and decline. As aforementioned in the instruction of the proposed framework (Sect. 2.3), therefore, the collaborative use-case scenario intends to elaborate on processes (*Process*) of ontology development and demonstrate working stages (*Stage*) of stakeholder's collaborative activities.

4.1 A Scenario of Scrum Modeling

As collaborative activities, the workshop of Semantic Web technology was organized, at Chiang Mai international convention and exhibition center, on February 2, 2019. This workshop aimed to motivate domain experts to develop domain ontologies, as a knowledge base for supporting organization activities.

In this workshop, ontology creators were domain experts in a cultural domain, and they had an objective to represent their knowledge as "Buddha Image" within a workshop duration (*Process 1*). As shown at the left side of Fig. 4, we introduced participants to understand a knowledge base and develop domain ontologies by using the ontology editor, called Hozo [15] (*Process 2*). The domain ontologies were designed based on a theory of role modeling.

In the third developing process (*Process 3*), as a source of knowledge, domain experts shared their knowledge in "Buddha Image" components. In the fourth process (*Process 4*), the ontology creator plan to design "Buddha Image" ontology and revise iteratively. As shown in the right side of Fig. 4, the ontology creator prepared the domain ontologies and presented to other participants, as *ontology construction backlog* (*Process 5*).

After that, as members of a cultural domain community, the workshop participants present the design of the domain ontology in a culture domain, as shown in the right side of Fig. 4 (*Process 6*).

The first version of the domain ontologies usually was not designed completely. To improve the domain ontologies, knowledge engineers gave comments to revise concepts and concept's properties, as *Sprint Review* (*Process 7*). As shown in Fig. 5, the "Buddha Image" ontology did not have appropriate concept properties. The "posture" concept has to be concept properties of "Buddha Image" concept.

However, the "Buddha Image" ontology needs to be discussed in many aspects of other participants (*Process 8*), such domain experts or knowledge engineers. Then, we encouraged the domain experts to register their account to the CD-OAM framework and upload the domain ontology, as ontology versioning (*Process 9*). We demonstrated how the framework can be used and how stakeholders can collaborate with their ontology design. We demonstrated how the

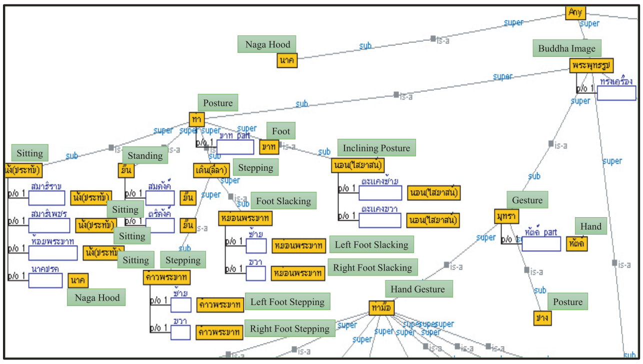


Fig. 5 Excerpt of "Buddha Image" ontology created by a culture domain expert using the ontology editor, Hozo [15], in the OAM framework [14]. Note that all Thai concept names (yellow rectangles) were translated into English (green rectangles).

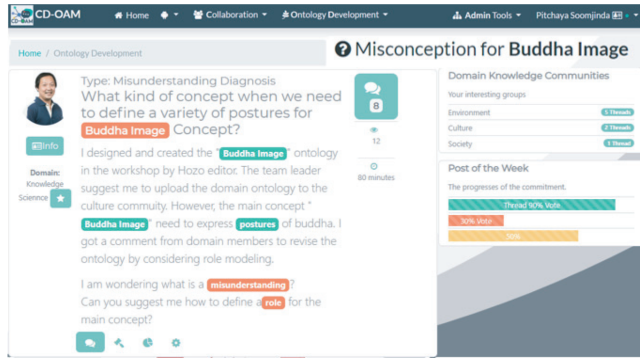


Fig. 7 Excerpt of a question of "misunderstanding diagnosis" in a module of knowledge-based management of the CD-OAM Framework [18]. A post of system users is notified to others in a discussion forum.

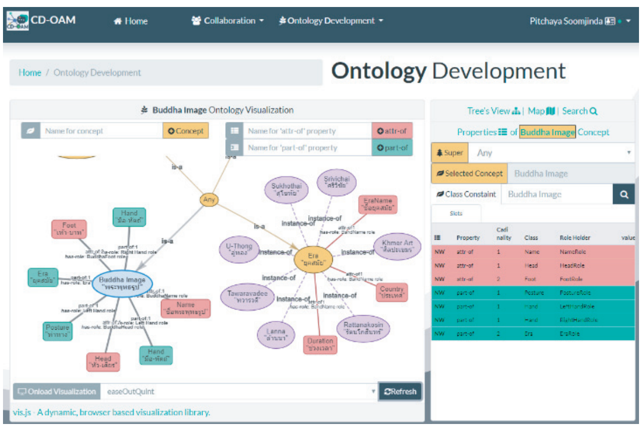


Fig. 6 Excerpt of "Buddha Image" ontology is visualized by the module of knowledge-based management of the CD-OAM Framework [18].

framework can be used and how stakeholders can collaborate with their ontology design. As illustrated in Fig. 6, the "Buddha Image" ontology was visualized by the module of knowledge-based management of the framework [16], as the last developing process (*Process 10*).

4.2 Roles and Collaborative Activities

As aforementioned in Sect. 2.3, Scrum model has been applied in the collaborative use-case scenario of the community of a cultural domain. Five types of stakeholders are categorized into the following roles: (1) the *ontology creator* is the domain expert from Chiang Mai University who first develop the "Buddha Image" ontology, (2) three *knowledge engineers* who give suggestions in ontology designing from NECTEC's research institute, (3) a group of *domain experts* in the culture domain who gave comments in other aspects about properties of concepts, (4) *team and experts* from other domains that they wanted to employ the "Buddha Image" ontology to explain their domain knowledge, such as a domain of tourism culture, and (5) *application users* who can use the simple application [18] from domain ontologies. When they log into the web applications based on the CD-OAM framework, the Scrum model was applied to define their working stages. Therefore, the five types of stakeholders with their roles can be supported by collaborative activities in ontology development.

4.3 Collaborative Framework Exploitations

The CD-OAM framework [18] provided the module of knowledge-based management for notifying relevant stakeholders. As mentioned in Sect. 4.1, the "Buddha Image" ontology needed to be reviewed in the *Sprint Cycle* (*Process 8*). System users of the community in a cultural domain were allowed to make comments or suggestions to the creator of "Buddha Image" ontology. Figure 7 illustrated the question type of "misunderstanding diagnosis" that is available to the relevant member for discussing in "Buddha Image" concepts. In this case, the domain expert from culture post a question about ontology designing. They needed to know "What kind of concepts when we need to define a variety of posture for Buddha Image Concepts". The system separates the types of the "Buddha Image" concept into two tagging colors that related to the terminologies of the different domain knowledge. Following the development phase in Scrum model, the framework provided a discussion forum that stakeholders can inquire to other users, such as domain experts or knowledge engineers.

5. Results and Discussion

This section explains a measurement result of our framework with related works, and discuss important issues.

We chose the *Capability Maturity Model Integration* (*CMMI*) [26] to measure related works of ontology development frameworks based on Agile and Scrum models. CMMI is a standard training and appraisal program to measure capability levels in two aspects: a developing process and a working stage. The following explanation discussed how to measure the related works in the three following steps.

In the first step, the CMMI guideline refers to a *process area* (*PA*) that is to cluster related practical applications for satisfying goals. As shown in Fig. 2, each phase of the workflow contains many keywords of Agile and Scrum models. To mapped keywords in Agile and Scrum Models with PAs, we preprocessed textual data of the guideline by extracting PA and its definition. We extracted those keywords by using *natural language processing* (*NLP*) [27]. Each preprocess requires different NLP techniques for data manipulation. We first removed the inflectional endings of each word by *tokenization* and filtered unneeded words by a *stop-words removing*. Then, we detected a sequence of adjacent keywords by *n-gram detection*. After that, we map the adjacent keywords of PL with the keywords based on Agile and Scrum models. Finally, we can map keywords with PAs of CMMI in different maturity levels. Table 2 shows the mapping result: Agile keywords (A1–A14) for collaborative activities with Maturity Level 3, and Scrum keywords (S1–S17) for working Stage with Maturity Level 2.

In the second step, we analyzed the way to characterize developing processes and working stages in their capabilities and performance by using a *capability level* (*CL*) of the guideline. As shown in Table 3, three numbers of *CL* are selected to identify and evaluate the methodologies based on Agile and Scrum models.

In the third step, the mapping result (the first step) and CL numbers (the second step) is used to measure the coverage of mapping Agile and Scrum models in term of the *process area coverage score* (*PACS*). In essence, PACS is the percentage of keywords that can be mapped from CMMI to Scrum/Agile. Suppose c is a capability level, and p is a process area. The PACS of the process area p is defined as:

$$PACS(p) = 100 \times \frac{\sum_{\acute{c} \in CA} \sum_{\acute{p} \in PA(\acute{C})} N_{\acute{c}\acute{p}}}{N_{cp} \times 2}$$

Where N_{cp} is the number of keywords that can be mapped from CMMI, CA is the set of all capability levels, and PA(c) is the set of the process areas in the capability level c.

As shown in Table 4, five related works and our work

Table 2 Mapped keywords in Agile and Scrum models with process areas of CMMI [26].

Capability Maturity Model Integration (CMMI)		Agile and Scrum Models
Agile Keyword (A) for Developing Process with Maturity Level 3		
Abbr.	Process Area	keywords
OPD	Organizational Process Definition	(A1) Domain of knowledge & Objective (A2) Tools & techniques (A3) Source selection
RD	Requirements Development	(A4) Sprint (A5) Sprint knowledge acquisition (A6) Sprint conceptualization (A7) Sprint formalization (A8) Sprint consensus (A9) Sprint evaluation
PI	Product Integration	(A10) Ontology construction planning (A11) Ontology construction backlog (A12) Ontology construction Sprint planning
IPM	Integrated Project Management	(A13) Sprint backlog (A14) Sprint review
Scrum Keyword (S) for Working Stage with Maturity Level 2		
Abbr.	Process Area	Keyword
REQM	Requirements Management	(S1) Knowledge Engineer (S2) Ontology Backlog (S3) Ontology Backlog Refinement
PP	Project Planning	(S4) Stakeholder (S5) Ontology Owner (S6) Domain Expert (S7) Team and Others (S8) Application User (S9) Ontology Creator
PMC	Project Monitoring and Control	(S10) Sprint (S11) Sprint Backlog (S12) Sprint Consensus Meeting (S13) Commitment during the Sprint cycle (S14) User Testing (S15) User Review
CM	Configuration Management	(S16) Ontology Versioning (S17) Members Notification

Table 3 Three numbers of capability levels (CL) of CMMI and meaning for evaluating methodologies based on Agile and Scrum models.

Capability Level (CL)	Meaning	Description
0	Incomplete	A process is not performed or only partially performed
1	Initial	A process satisfies the specific goals of a process area enabling work products to be produced
2	Managed	A collaborative framework is in place to support the process

Table 4 A comparison table of related works considering keywords in Agile and Scrum models with CMMI standard [26].

Agile Keyword (A) for Developing Process with Maturity Level 3		Capability Level of keyword of mapping Agile and Scrum models with CMMI					
Process Area	Keyword	XP.K [9]	Rapid OWL [10]	AOSD [11]	SAMOD [12]	AMOD [13]	CD-OAM [18]
OPD	A1	2	1	2	2	2	2
	A2	2	2	2	1	2	2
	A3	1	1	1	1	2	2
RD	A4	0	0	1	0	2	2
	A5	0	0	0	0	2	1
	A6	0	0	0	0	2	2
	A7	0	0	0	0	2	1
	A8	0	0	0	0	0	2
	A9	0	0	0	0	2	2
PI	A10	2	1	0	0	1	2
	A11	0	0	0	0	1	2
	A12	0	0	0	0	1	2
IPM	A13	0	0	0	1	2	1
	A14	0	0	0	0	2	2
Scrum Keyword (S) for Working Stage with Maturity Level 2		**Capability Level of keyword of mapping Agile and Scrum models with CMMI**					
Process Area	**Keyword**	XP.K [9]	Rapid OWL [10]	AOSD [11]	SAMOD [12]	AMOD [13]	CD-OAM [18]
REQM	S1	2	2	0	2	2	2
	S2	0	0	0	0	2	1
	S3	0	0	0	0	2	1
PP	S4	2	0	2	2	2	2
	S5	2	0	1	2	2	2
	S6	2	2	0	2	2	2
	S7	2	2	2	1	2	2
	S8	2	2	2	0	0	1
PMC	S9	0	0	2	2	2	2
	S10	0	0	2	0	2	2
	S11	1	0	1	0	2	2
	S12	0	0	0	0	2	2
	S13	0	0	0	2	0	2
	S14	2	1	2	2	2	1
	S15	2	1	2	0	2	1
CM	S16	2	2	0	2	**1**	2
	S17	0	0	0	0	**0**	2

were measured by the PACS method, including XP.K [9], RapidOWL [10], AOSD [11], SAMOD [12], AMOD [13], and CD-OAM [18]. Each keyword in the mapping result was identified by the CL number regarding their purposed methodologies and frameworks.

To elucidate how we assess the related works, we give an explanation in the following example. The AMOD work [13] mentioned: "*Configuration Management (CM): The goal of this activity is to record all the versions of the documentation and ontology code and to control the changes. Generally, CM of ontology includes four activities derived from the software engineering: configuration identification, configuration control, and configuration audits*".

Following how to measure the related works, the first step is to analyze PA (Table 2) that the AMOD initialized ontology versioning (S16) and did not perform members notification (S17). The second step is to define the CL numbers (Table 3): 1 as "Initial" for S16, and "Incomplete" as 0 for S17. In the third step, the assessed result of the CM shows CL's scores 1 and 0 for S16 and S17, respectively. Lastly, the PACS score of the CM (Table 5) is 25.0 and we can calculate by the PACS model as below:

$$PACS(CM) = 100 \times \frac{(0 \times CL2) + (1 \times CL1) + (0 \times CL0)}{(2PA) \times 2}$$
$$= 100 \times \frac{(0 \times 2) + (1 \times 1) + (0 \times 0)}{2 \times 2}$$
$$= 100 \times \frac{1}{4}$$
$$= 25$$

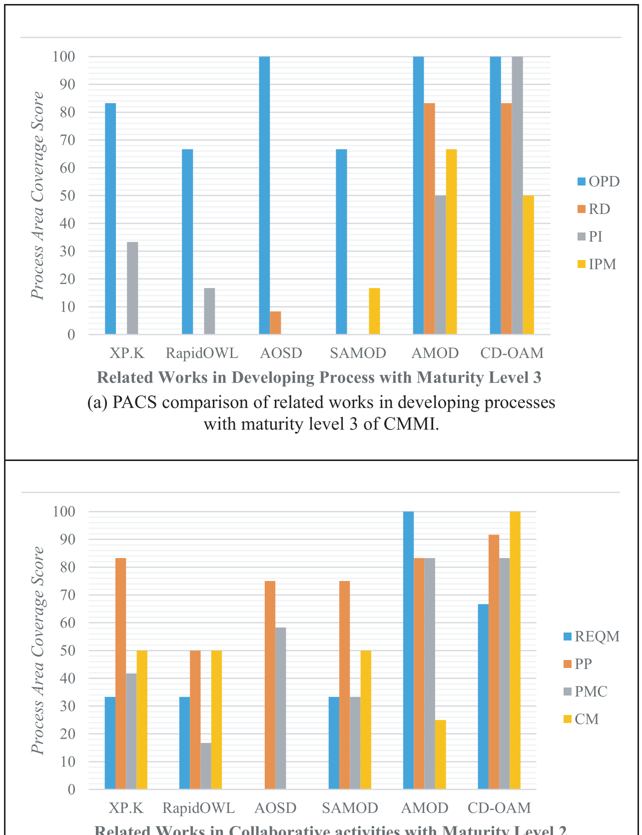
We followed the three steps of the measurement. The PACS method was used to calculate the capabilities and performance of the five related works and our work, and Table 5 shows all the measurement results. After comparing and analyzing the result, we found that all the works have a high score in Organizational Process Definition. AMOD and CD-OAM are the only two works that can present a prominent of Agile and Scrum models.

In addition, Fig. 8 contains two bar graphs that compare the measurement results of all the works in the two aspects: developing processes, and working stages. First, the top part (a) of Fig. 8 shows that AMOD and CD-OAM provided developing processes with maturity level 3 of CMMI covering all four process areas: OPD (Organizational Process Definition), RD (Requirements Development), PI (Product Integration), and IPM (Integrated Project Management). Second, the bottom part (b) of Fig. 8 shows that AMOD has the highest PACS score in REQM (Requirements Management), and CD-OAM has the highest PACS score in CM (Configuration Management). Therefore, the methodology based on Agile and Scum models can support the CD-OAM framework to work with ontology development and also organize stakeholders effectively working.

As demonstrated in the collaborative use-case scenario, the CD-OAM framework supported the development of the "Buddha Image" ontology collaboratively. Moreover, the improved framework can support the development of multidisciplinary ontologies that allows different domain stakeholders to share their understanding of a single discipline to cover relevant domains. For example, the stakeholders in a tourism domain can develop their "Historical travel" ontology and share the "Era" concept of the "Buddha image" ontology to describe the "historical location" concept of the "Historical travel" ontology that is one of the tourism domains.

Table 5 A comparison table of related works in the process area coverage score (PACS) for Agile and Scrum models with CMMI [26].

Process Area	XP.K [9]	RapidOWL [10]	AOSD [11]	SAMOD [12]	AMOD [13]	CD-OAM [18]
	Process area coverage score for Agile model with maturity level 3					
OPD	83.3	66.7	100.0	66.7	100.0	100.0
RD	0.0	0.0	8.3	0.0	83.3	83.3
PI	33.3	16.7	0.0	0.0	50.0	100.0
IPM	0.0	0.0	0.0	16.7	66.7	50.0
	Process area coverage score for Scrum model with maturity level 2					
Process Area	XP.K [9]	RapidOWL [10]	AOSD [11]	SAMOD [12]	AMOD [13]	CD-OAM [18]
REQM	33.3	33.3	0.0	33.3	100.0	66.7
PP	83.3	50.0	75.0	75.0	83.3	91.7
PMC	41.7	16.7	58.3	33.3	83.3	83.3
CM	50.0	50.0	0.0	50.0	25.0	100.0



(a) PACS comparison of related works in developing processes with maturity level 3 of CMMI.

(b) PACS comparison of related works in working stages with maturity level 2 of CMMI.

Fig. 8 Two comparing bar graphs of related works in the measurement of the process area coverage score (PACS) [26]: (a) PACS comparison for related works in developing processes with maturity level 3 of CMMI, and (b) PACS comparison for related works in working stages with maturity level 2 of CMMI.

In the findings, we are aware that there are some limitations. In particular, the stakeholders involved in the ontology development have a common ground of Buddhism belief. This simplifies the process of knowledge sharing, because the majority of Buddhism terminology are understood by all of them.

However, there are certain cases where common terms are less shared among multiple disciplines. For example, the management of COVID-19 pandemic which involves healthcare, economics, transportation, education, and society, where multidisciplinary precautions and concerns have to be taken into account. Misunderstanding in these cases is much deeper than the word level and, therefore, out of scope of this paper.

To alleviate this issue, we envisage a collaborative framework that is capable of reconciling these conflicts via the power of knowledge graph. First, the system should allow stakeholders to express their precautions and concerns with freedom. It then matches them across domains, classify them into agreement and disagreement, and list them up in a common table. Next, the system displays this table and commences an open discussion based on the conflicts. The system facilitates the discussion by associating the conflicts with the Network Text Analysis technique and identifies which domains they are related to. Finally, once the experts settle all conflicts, the system summarizes the results as a knowledge graph.

6. Conclusion

This paper presents the enhancement of a community-driven ontology-based application management (CD-OAM) framework by employing Agile and Scrum models to systematizing collaborative activities. The CD-OAM framework was improved in parts of developing processes of ontology development and working stages of stakeholder's collaboration. The measurement results pointed out that Agile and Scum models can work together effectively.

Our main contributions were concluded as follows. First, the proposed processes based on Agile and Scrum models described processes of ontology development following standard guidelines and collaborative ontology development. Second, the proposed collaborative framework based on Agile and Scrum models provided system features for supporting ontology development. The collaborative use-case scenario demonstrated the framework's features supported the stakeholders in a cultural domain in two aspects. The first aspect is developing processes of the "Buddha Image" ontology collaboratively, and the second aspect is to facilitate working stages for stakeholder's discussion in the creation of cross-domain concepts/relations of multidisciplinary ontologies.

We can give an outlook on further work with the knowledge graph that the framework can be employed to reduce miscommunication in cross-disciplinary concept discovery through network text analysis and semantic embedding.

Acknowledgments

The authors are special thanks to reviewers for fruitful technical comments. This research is partially supported by National Electronics and Computer Technology Center (NECTEC), Thailand.

References

[1] I. Horrocks, "Ontologies and the semantic web," Commun. ACM, vol.51, no.12, pp.58–67, 2008.
[2] G. Stephan, H. Pascal, and A. Andreas, "Knowledge representation and ontologies," Semant. Web Serv., Concepts, Technol. Appl., pp.51–105, Springer, 2007.
[3] R. Mizoguchi and K. Kozaki, "Ontology engineering environments," Handbook on Ontologies, pp.315–336, Springer, 2009.
[4] T. Slimani, "Ontology development: A comparing study on tools, languages and formalisms," Indian J. Sci. Technol., vol.8, no.24, pp.1–12, 2015.
[5] K.W. Collier, Agile Analytics: A Value-Driven Approach to Business Intelligence and Data Warehousing: Delivering the Promise of Business Intelligence (Agile Software Development), Addison-Wesley, 2011.
[6] K. Schwaber, "SCRUM development process," Business Object Design and Implementation, pp.117–134, Springer, 1997.
[7] D. Gašević, N. Kaviani, and M. Milanović, "Ontologies and software engineering," Handbook on Ontologies, pp.593–615, Springer, 2009.
[8] Y. Sure, M. Erdmann, J. Angele, S. Staab, R. Studer, and D. Wenke, "OntoEdit: Collaborative ontology development for the semantic web," International Semantic Web Conference, Lecture Notes in Computer Science, vol.2342, pp.221–235, Springer, 2002.
[9] H. Knublauch, "Ontology-driven software development in the context of the semantic web: An example scenario with Protege/OWL," 1st International Workshop on the Model-Driven Semantic Web (MDSW2004), pp.381–401, 2004.
[10] S. Auer and H. Herre, "RapidOWL—An agile knowledge engineering methodology," International Andrei Ershov Memorial Conference on Perspectives of System Informatics, Lecture Notes in Computer Science, vol.4378, pp.424–430, Springer, 2006.
[11] D. Parsons, "An ontology of agile aspect oriented software development," Res. Lett. Inf. Math. Sci., vol.15, pp.1–11, Nov. 2011.
[12] S. Peroni, "A simplified agile methodology for ontology development," OWL: Experiences and Directions—Reasoner Evaluation, Lecture Notes in Computer Science, vol.10161, pp.55–69, Springer, 2017.
[13] A.S. Abdelghany, N.R. Darwish, and H.A. Hefni, "An agile methodology for ontology development," Int. J. Intell. Eng. Syst., vol.12, no.2, pp.170–181, 2019.
[14] M. Buranarach, T. Supnithi, Y.M. Thein, T. Ruangrajitpakorn, T. Rattanasawad, K. Wongpatikaseree, A.O. Lim, Y. Tan, and A. Assawamakin, "OAM: An ontology application management framework for simplifying ontology-based semantic web application development," Int. J. Softw. Eng. Knowl. Eng., vol.26, no.1, pp.115–145, 2016.
[15] K. Kozaki, Y. Kitamura, M. Ikeda, and R. Mizoguchi, "Hozo: An environment for building/using ontologies based on a fundamental consideration of "Role" and "Relationship"," Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web, Lecture Notes in Computer Science, vol.2473, pp.213–218, Springer, 2002.
[16] T. Tudorache, C. Nyulas, N.F. Noy, and M.A. Musen, "WebProtégé: A collaborative ontology editor and knowledge acquisition tool for the web," Semant. Web, vol.4, no.1, pp.89–99, 2013.

[17] K. Kozaki, E. Sunagawa, Y. Kitamura, and R. Mizoguchi, "A framework for cooperative ontology construction based on dependency management of modules," Proc. First International Conference on Emergent Semantics and Ontology Evolution, pp.33–44, 2007.
[18] A. Takhom, S. Usanavasin, T. Supnithi, and M. Ikeda, "Collaborative ontology development approach for multidisciplinary knowledge: A scenario-based knowledge construction system in life cycle assessment," IEICE Trans. Inf. Syst., vol.E101-D, no.4, pp.892–900, April 2018.
[19] C.K.C. Bernard and W.P.P. Anita, " Multidisciplinarity interdisciplinarity and transdis ciplinarity in health research services education and policy: 1. Definitions, objectives, and evidence of effectiveness," Clin. Investig. Med., vol.29, no.6, pp.351–364, 2006.
[20] J. Cho, "Issues and challenges of agile software development with SCRUM," Issues Inf. Syst., vol.9, no.2, pp.188–195, 2008.
[21] T. Otwell, "Collections - Laravel - The PHP Framework For Web Artisans," http://laravel.com/. p.1, 2015.
[22] L. Welling and L. Thomson, PHP and MySQL Web Development, Sams Publishing, 2003.
[23] G.E. Krasner and S.T. Pope, "A description of the model-view-controller user interface paradigm in the smalltalk-80 system," J. Object Oriented Program., vol.1, no.3, pp.26–49, 1988.
[24] N.F. Noy and D.L. McGuinness, "Ontology development 101: A guide to creating your first ontology," Stanford Knowledge Systems Laboratory, Stanford knowledge systems laboratory technical report KSL-01-05 and Stanford medical informatics technical report SMI-2001-0880, Stanford, CA, p.25, 2001.
[25] A. Takhom, P. Boonkwan, M. Ikeda, S. Usanavasin, and T. Supnithi, "Reducing miscommunication in cross-disciplinary concept discovery using network text analysis and semantic embedding," The 6th Joint International Semantic Technology Conference, CEUR Workshop Proceedings 1741, vol.2000, pp.20–31, 2017.
[26] M.B. Chrissis, M. Konrad, and S. Shrum, CMMI - Guidelines for Process Integration and Product Development, Addison-Wesley Longman Publishing, 2003.
[27] S. Bird, E. Klein, and E.L. Beijing, Natural Language Processing with Python, 1st ed., O'Reilly Media, 2009.

Akkharawoot Takhom received the B.S. degree in Management of Information Technology from Mae Fah Luang University in 2009, and M.Eng. degree in Information and Communication Technology for Embedded Systems from SIIT in 2013. He received a Ph.D. degree in Knowledge Science from the JAIST in 2018.

Sasiporn Usanavasin received B.Sc. degree in Information Technology from SIIT in 1999 M.Eng. degree in Software Engineering from Keio University in 2003. She received a Ph.D. in Software Engineering, Keio University, Japan in 2006. She is now with the School of Information, Computer, and Communication Technology, SIIT, Thailand.

Thepchai Supnithi received the B.S. degree in Mathematics from Chulalongkorn University in 1992. He received the M.S. and Ph.D. degrees in Engineering from the Osaka University in 1997 and 2001, respectively. Since 2001, he has been with the Human Language Technology Laboratory, NECTEC, Thailand.

Prachya Boonkwan received B.Eng. and M.Eng. degrees in Computer Engineering from Kasetsart University in 2002 and 2005, respectively. He received a Ph.D. degree in Informatics from the University of Edinburgh, UK, in 2014. Since 2005, he has been with Language- and Semantic Technology Lab at NECTEC in Thailand.