Assessing the Completeness and Clarity of UML for Conceptual Modeling

Prabodha Tilakaratna and Jayantha Rajapakse

***Abstract*—Unified Modeling Language (UML) is a popular modeling grammar which is widely being used in the information system modeling process. Although initially UML more focused on system modeling, recent research studies have identified that UML can also be used in conceptual modeling. Therefore, this paper quantitatively assesses the appropriateness of UML for conceptual modeling using two significant factors: (1) to what extent the constructs of UML can represent the real world domain characteristics (completeness) (2) whether UML can represent the real world domain characteristics uniquely using its constructs (clarity). The results indicate that, UML is not capable of achieving a total completeness and clarity. Therefore, the paper recommends that UML needs to be refined in a way to achieve a higher completeness and clarity, to be used for conceptual modeling.**

***Index Terms*—Completeness and clarity, conceptual modeling, ontology, unified modeling language (UML).**

I. Introduction

Accurate development of an information system involves modeling the information system before moving to the implementation [1]. Information system modeling primarily carries out in two phases: (1) modeling the characteristics of a real world domain, which needs to be automated as an information system (conceptual modeling) (2) modeling the characteristics of the information system to be developed (system modeling).

Few decades earlier, in both conceptual and system modeling, the data and process characteristics were captured and modeled separately [2]. Data characteristics were modeled using modeling grammars such as ERM (Entity Relationship Modeling) [3] and process characteristics were modeled using grammars such as DFD (Data Flow Diagrams), workflow specification grammars [4], [5] and business process modeling grammars [6]. Software developers and researchers have identified that the characteristics of real world domains and information system cannot be precisely represented with this distinction. Therefore, they came up with a new information system development concept namely object-oriented concept, which centralized both the data and process characteristics [7]-[8]. With the rapid emergence of object-oriented concept, various grammars evolved for both object-oriented information system modeling and implementation.

Unified Modeling Language (UML) is one such popular modeling grammar, which was developed based on the object-oriented concept [9]. UML is developed in a way where it is intended to be used with all information system development methodologies which supports object-oriented concepts. Also UML can be used with any software development life cycles, and support most of the object-oriented development processes [10]-[11]. UML has thirteen different types of diagrams which have the capability to accurately model almost all the characteristics of an information system [10]. In addition, UML diagrams can easily be transformed into program codes during the information systems implementation [12]. Moreover, although UML is defined as a precise modeling grammar, it is not a barrier to include further improvements in UML concepts [9]. Because of the aforementioned factors, UML is widely being used in the current information system development field.

During the time of the origination of UML, its' grammatical constructs and rules were defined by assigning the characteristics of information systems [13]. Therefore, UML was more appropriate to be used in system modeling. However, recently some researchers refined UML by assigning the characteristics of the real world domains into UML constructs [13]-[14]. Based on the results obtained, the researchers stated that UML constructs are capable of accurately representing the real world domain characteristics in conceptual models [13]-[16].

However, the characteristics of real world domains can be divided into three main categories: (1) static characteristics (2) state changes of the static characteristics and (3) interactions between the static characteristics. Above research studies only considered the (1) and (2) categories, during the refinement of UML [9]-[10]. Therefore, it cannot be specified that the refined UML includes necessary constructs to represent the real world domain characteristics, which belongs to *all* the three categories (i.e. *completeness* of UML). Moreover, no previous research study carried out to assign the real world domain characteristics of all the three categories, with the UML constructs.

In addition, none of the previous studies assessed whether each real world domain characteristic can be assigned into only a single UML construct, by having a one-to-one assignment between the real world domain characteristics and UML constructs (i.e. *clarity* of UML). Thus, the aim of this paper is to assess the suitability of UML for conceptual modeling, based on the completeness and clarity. We use the representational theory proposed by Ron Weber [17], and its' associated concepts of completeness and clarity as the

Manuscript received September 19, 2012; revised October 31, 2012.

Prabodha Tilakaratna is with the Monash University Sunway Campus, Jalan lagoon selatan, Bandar sunway, 46150, Selangor Darul ehsan, Malaysia (e-mail: prabodha@ monash.edu).

Jayantha Rajapakse is with the Monash University Sunway Campus, Jalan lagoon selatan, Bandar sunway, 46150, Selangor Darul ehsan, Malaysia (e-mail: jayantha.rajapakse@monash.edu).

DOI: 10.7763/IJIMT.2012.V3.331

measurements of this study. From the aforementioned theory and concepts, the following two factors have been identified and used in assessing UML for conceptual modeling:

1) To what extent UML is capable of representing the real world domain characteristics using its grammatical constructs?
2) Whether UML can represent each real world domain characteristic uniquely using its grammatical constructs?

Factor (1) and (2) respectively measure the *completeness* and *clarity* of the modeling grammars being used in the conceptual modeling process [17]. Hence, in this study, the appropriateness of UML is being assessed using its completeness and clarity.

The remainder of this paper is structured as follows. Section 2 provides a description about Weber's representational theory [17] including the previous applications of this theory for modeling grammars. Findings and explanations related to the results obtained regarding the appropriateness of UML are given in Section 3. Finally, the paper concludes with a discussion of the overall research implications and future research directions in Section 4.

II. Background and Related Work

Although different object-oriented modeling grammars emerged with the introduction of object-oriented concept, all of them do not possess necessary constructs and rules to represent the real world domain characteristics completely and clearly [18]. Use of inappropriate modeling grammars for conceptual modeling process will create erroneous modeling scripts, which ultimately leads to erroneous information systems [19]. Therefore, researchers have identified the requirement of assessing the appropriateness of modeling grammars before using them for the conceptual modeling process. The investigations performed on this requirement resulted in the introduction of a new theory namely, the *ontological expressiveness theory* [20].

This theory assesses the strengths and weaknesses of information system development methodologies in expressing the real world domain characteristics seamlessly. While developing the theory, those researchers focused on evaluating the expressiveness of information system development methodologies based on three main factors:

1) The capability to represent the real world domain characteristics.
2) The capability to track the state changes of real world domains.
3) The capability to decompose the conceptual models developed from the real world domains.

Out of the three factors, in order to assess the representational capability (i.e. first factor), Wand and Weber (the two researchers who developed the ontological expressiveness theory) created a model during their study which is known as *representational model*. This model was used to assess how clearly and completely an information system represents the characteristics of the real world domains, using the underlying modeling grammars.

Weber further analyzed the ontological expressiveness theory and the representational model, and came up with an extension to the ontological expressiveness theory namely, *representational theory* [17]. The core concept of this theory involves the use of modeling grammars to represent the real world domain characteristics seamlessly in the conceptual modeling process. Representing the real world domain characteristics comprises discussing what exists in the real world domains and how they exist. During the development of the ontological expressiveness theory, Wand and Weber identified what exist in the real world domains, how things behave in the real world domains and how the characteristics of real world domains can comprehensively be observed using philosophical disciplines [21].

Ontology is a subdivision of philosophy, which signifies the abstraction and representation of different domain characteristics. Thus, ontological approach was used in developing both the ontological expressiveness theory and the representational theory. Mario Bunge's ontology [22], [23] has been chosen for the development of the two theories, because this was the only existing ontology which covered the entire real world by discussing its domain characteristics and the relationships between those characteristics. Besides, the most widely used ontology in the current information systems research field was that of Bunge [15–17], [20]. Therefore, using this ontology, Wand and Weber derived a set of ontological concepts and they argued that this set of concepts has the capability to cover the real world domain characteristics in general.

Hence, these concepts been used in the representational theory to assess the completeness and clarity of modeling grammars in representing the real world domain characteristics. This is known as the *representational capability* of a modeling grammar [17]. Weber defined that, *completeness* means the capability of a modeling grammar to represent any given real world domain scenario without losing information. Conversely, *clarity* means the ability to uniquely represent the real world domain characteristics using the grammatical constructs. Completeness and clarity of modeling grammars were assessed by mapping the grammatical constructs with the previously defined set of ontological concepts [20]. This mapping was performed adhering to the mapping requirements proposed by Wand and Weber in the ontological expressiveness theory [20].

According to these requirements, the mapping can be bi-directional, but it should ensure one-to-one relationships between the two sets being used. However, Weber has figured out that except one-to-one relationships, certain other mapping relationships can also exist between the grammatical constructs and the ontological concepts. Such relationships were identified as incorrect relationships and were named as representational *deficiencies.* The identified representational deficiencies could be divided into four main categories as construct deficit, redundancy, overload and excess.

Weber has observed that, construct deficit affects and undermines the completeness of modeling grammars [17]. *Construct deficit* means one-to-zero relationships, where certain ontological concepts do not have matching grammatical constructs. Furthermore, Weber stated that the remaining representational deficiencies (i.e. construct redundancy, overload and excess) effect on the clarity of modeling grammars. *Construct redundancy* means one-to-many relationships, where one ontological concept can has more than one matching grammatical constructs. *Construct overload* means many-to-one relationship, where more than one ontological concept has only one matching grammatical construct. *Constructs excess* means zero-to-many relationship, where certain grammatical constructs do not have matching ontological concepts.

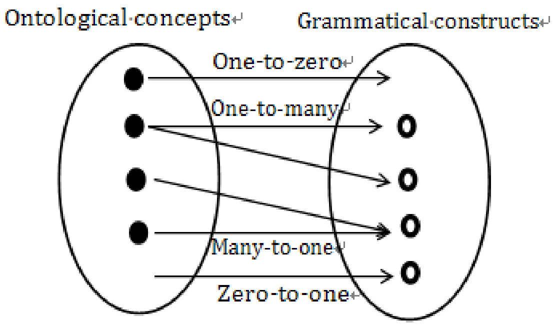

Fig. 1. Four representational deficiencies

Subsequently, the representational theory along with the completeness and clarity has been applied in the conceptual modeling area successfully, by many researchers as follows.

A set of researchers conducted investigations by utilizing the representational theory in the e-Business area [24], [25]. They used this theory to examine the e-Business in terms of completeness and clarity by using its underlying tool, which resulted in a higher ontological completeness [25]. Furthermore, the representational theory was used to compare different modeling techniques that were being used in the enterprise system interoperability [24]. Four modeling techniques were used for the comparison and the ontological completeness and clarity of those techniques were assessed using the theory.

Another research study carried out to assess the representational capability of modeling grammars being used in process modeling [6]. Representational theory was used to assess twelve selected process modeling grammars in representing the real world domain characteristics. Based on the analysis they have identified the process modeling grammars which are most suitable to be used in the business process modeling area. Keen and Lakos are two researchers who used representational theory to assess the completeness of six process modeling techniques using some essential features identified by them [26]. Based on the analysis, they concluded that, representational theory can effectively be used in interpreting and comparing the process modeling techniques. Green and Rosemann examined the completeness and clarity of Event-Driven Process Chain (EPC) notation using the representational theory [27].

Some recent research studies focused on utilizing the set of ontological concepts proposed by Wand and Weber [20] to assess the applicability of UML in the conceptual modeling process. Nevertheless, they did not focus on assessing the completeness or clarity of UML in representing the real world domain characteristics by referring the representational theory [17]. Therefore, this area still remains undiscovered, where further investigations can be carried out.

III. Representational Capability Assessment

This section exemplifies the investigation with regards to the assessment of the representational capability (completeness and clarity) of UML, which carried out based on the representational theory.

A. Completeness of UML

As specified in the previous section, completeness of UML will be assessed using the *construct deficit*. In simple terms, if construct deficit exists with a particular modeling grammar, some of the real world domain characteristics would not be able to represent from the modeling script developed using that grammar. To identify whether UML is not capable of representing certain real world domain characteristic, UML constructs have been mapped with the ontological concepts and the results are given in Table I. Each tick appeared in the table indicates that the particular ontological concept can be represented by a construct of UML. Based on the mapping, the total degree of deficit calculated considering all the unmapped ontological concepts relative to the total ontological concepts as follows:

$$\text{Total degree of deficit} = \frac{\text{Number of ontological concepts that do not have matching UML construct}}{\text{Total number of ontological concepts}}$$

Since the construct deficit illustrates the incompleteness of a modeling grammar, total degree of completeness has been calculated as follows:

Total degree of completeness = (1 – Total degree of deficit) %

According to the above equations, the completeness of UML or the ability of UML in representing any real world domain characteristic, has achieved 68.42%. Conversely, from 31.58%, UML do not contain necessary constructs to represent some real world domain characteristics.

B. Clarity of UML

The main requirement of clarity involves, the unique representation of real world domain characteristics using each UML construct. The remaining three representational deficiencies i.e. construct redundancy, overload and excess, will be considered in this sub section for the assessment of the clarity of UML. Firstly, we consider construct redundancy. Redundant constructs mean having more than one grammatical construct to represent the same real world domain characteristic, which may create confusions during the modeling process. The UML constructs have been mapped with the ontological concepts as depicted in Table II, and we have identified the numbers of UML constructs that can be mapped with each ontological concept. The total degree of redundancy was calculated considering all the UML constructs which represented one ontological concept using more than one UML construct as follows.

TABLE I: ASSESSING THE COMPLETENESS OF UML USING CONSTRUCTS DEFICIT

Ontological concept (ontological *thing* and related concepts)	**UML**	**Ontological concept (ontological *state* and related concepts)**	**UML**	**Ontological concept (ontological *event* and related concepts)**	**UML**	**Ontological concept (ontological system and related concepts)**	**UML**
Thing	√	State	√	Event	√	System	√
Class		Conceivable state space		Conceivable event space	√	System composition	√
Natural kind	√	State law	√	Lawful event space	√	System environment	√
Property	√	Lawful state space		External event	√	System structure	√
Intrinsic property	√	Stable state		Internal event	√	Sub system	
Mutual property	√	Unstable state		Well defined event		System decomposition	√
Emergent property	√	History		Poorly defined event		Level structure	
State function		Action	√	Transformation	√		
Functional schema	√			Lawful transformation	√		
Quantitative change	√			Coupling	√		
Qualitative change				Process	√		
Interaction	√						
Degree of deficit	25.00 %	Degree of deficit	62.50 %	Degree of deficit	18.18 %	Degree of deficit	28.57 %
The degree of deficit							**31.58%**
The degree of completeness							**68.42%**

TABLE II: ASSESSING THE CLARITY OF UML USING CONSTRUCTS REDUNDANCY

Ontological Concept	**UML**	**Ontological Concept**	**UML**	**Ontological Concept**	**UML**	**Ontological Concept**	**UML**
Thing	2	State	4	Event	4	System	2
Class	0	Conceivable state space	0	Conceivable event space	2	System composition	2
Natural kind	2	State law	1	Lawful event space	2	System environment	1
Property	1	Lawful state space	0	External event	1	System structure	1
Intrinsic property	1	Stable state	0	Internal event	1	Sub system	0
Mutual property	1	Unstable state	0	Well defined event	0	System decomposition	1
Emergent property	1	History	0	Poorly defined event	0	Level structure	0
State function	0	Action	3	Transformation	2		
Functional schema	2			Lawful transformation	2		
Quantitative change	1			Coupling	1		
Qualitative change	0			Process	2		
Interaction	3						
The degree of redundancy							**43.48%**

$$\text{Total degree of redundency} = \frac{\text{Number of UML constructs that have a mapping to the same ontological concept}}{\text{Total number of UML constructs}}$$

Next, we have moved to construct excess. Existence of modeling grammar constructs which do not have any meaningful characteristic to be mapped in the intended domain will create construct excess. Excess UML constructs means the constructs of UML which cannot be mapped with any real world domain characteristic. Unavailability of matching real world domain characteristics will make such constructs worthless in the conceptual modeling process.

Therefore, excess UML constructs will not be utilized in creating conceptual models. After mapping the UML constructs with the ontological concepts, the excess UML constructs have been identified which are given in Table III. The total degree of excess was calculated by considering all the excess UML constructs relative to the total UML constructs as follows:

$$\text{Total degree of excess} = \frac{\text{Number of UML constructs that do not have a mapping to any of the ontological concept}}{\text{Total number of UML constructs}}$$

The final representational deficiency which used to assess the UML clarity is *construct overload.* This illustrates the situation where one UML construct is used to represent multiple real world domain characteristics. This can create difficulties during the development of conceptual models, because different modelers may use different UML construct to represent the same real world domain characteristic. Therefore, it may be difficult for the stakeholders to grasp the accurate idea given by the conceptual model. From the mapping between the UML constructs and the ontological concepts, the UML constructs have been identified, which can be mapped with more than one ontological concept (Table IV). The total degree of overload has been calculated considering all the UML constructs that have multiple matching real world domain characteristics, relative to the total UML constructs:

$$\text{Total degree of overload} = \frac{\text{Number of UML constructs that can be mapped with more than one ontological concept}}{\text{Total number of UML constructs}}$$

TABLE III: ASSESSING THE CLARITY OF UML USING CONSTRUCTS EXCESS

Grammar	Proposed excess grammatical constructs	Degree of excess
UML	Classifier, Enumeration, Qualifier, Constraint, Use case inheritance relationship between actors, Composite state, Guard, Destroy object, Combined fragment, Class role, Synchronization	23.91%

In order to obtain a single value for the total degree of clarity, the numerical values obtained for the three representational deficiencies need to be merged together. Nevertheless, this merging requires the existence of a common source, thereby relative to the common source, the values of the three representational deficiencies can be added together. The ability of UML to represent any given real world domain characteristic (i.e. completeness) has been considered as the common source, and the values of the three representational deficiencies relative to the total degree of completeness were identified (Table V). Since the three representational deficiencies decrease the clarity, those relative values were used to achieve a single value for the lack-of-clarity of UML. Hence, the final value obtained for the *total degree of clarity* of UML involves 62.93%

TABLE IV: ASSESSING THE CLARITY OF UML USING CONSTRUCTS OVERLOAD

Grammar	Proposed overload grammatical constructs	Degree of overload
UML	Use case relationship Activity, Activation, Event	8.70%

C. Discussion

Based on the results obtained in the previous two sub sections, it can be identified that the conceptual models create using UML will not be hundred percent accurate due to its lack of total completeness and clarity. Therefore, refining the UML constructs and rules is required to achieve a total representational capability.

To achieve the remaining 31.58% completeness, new or modified UML constructs need to identify to be mapped with the remaining unmapped real world domain characteristics. Similarly, to remove the lack-of-clarity, UML should be refined in a way to remove one-to-many, many-to-one and zero-to-one relationships that exist between UML constructs and ontological concepts. Nevertheless, refining an existing modeling grammar is a process which needs to be done in a careful manner, since it would be worthless if the refinements make the grammar complex. Also, all the refinements should be done in a way where such modifications can clearly be justified over the existing grammatical constructs and rules.

IV. IMPLICATIONS AND FUTURE WORKS

This research study has significant implications in both the practice and research as follows. When we consider the practical aspect, the research outcomes are significant for all the stakeholders of the conceptual modeling process such as modelers, designers and users. This is for the reason that, in terms of completeness and clarity, modelers and designers can have a clear idea about the extent to which UML is capable of representing the real world domain characteristics. This will help them to decide whether to use this modeling grammar for the modeling process or not. Moreover, although UML will be utilized, the stakeholders would be able to identify that, achieving hundred percent accurate conceptual models is impossible, due to the lack of completeness and clarity of UML in representing the real world domain characteristics. Also, knowing the completeness and clarity of UML will help them in dealing with the ambiguity situations that occur during the modeling process as well as in viewing the completed conceptual models.

If we consider the results from the research point of view, researchers can compare the representational capability of UML with the other research studies performed on the representational capability of certain modeling grammars [6], [26]. This will help them to have a clear idea about the level of the appropriateness of UML with regards to the other modeling grammars being used in the conceptual modeling process. Also, the same approach which has been used in this paper can be applied to other object-oriented modeling grammars to assess their representational capability. Most significantly, this study can be considered as another research work which proved the validity and the effective use of representational theory in assessing the modeling grammars for conceptual modeling.

This research study will further be expanded with the investigations of refining UML constructs and rules for increasing the completeness and clarity. Modifications required to be carried out for the UML constructs and rules until it reaches a total completeness and clarity. Subsequently, an empirical study will be performed to assess the success and the usability of the refined UML in the practical scenario.

TABLE V: ASSESSING THE TOTAL DEGREE OF LACK-OF-CLARITY OF UML

Relative value of the representational deficiency	Calculation	Final value
Degree of construct redundancy, relative to the degree of completeness	(43.48 / 68.42) %	63.55 %
Degree of construct excess, relative to the degree of completeness	(23.91 / 68.42) %	34.95 %
Degree of construct overload, relative to the degree of completeness	(8.70 / 68.42) %	12.72 %
Total degree of lack-of-clarity	(63.55 + 34.95 + 12.72) / 3	37.07 %

This research study has some limitations as follows. Firstly, the real world domains consist of an infinite number of characteristics and hence it is impossible to map all these real world domain characteristics with UML constructs. Therefore, we utilized the set of ontological concepts proposed by Wand and Weber [20], and as they stated we assume that those concepts can represent the real world domains in general. Secondly, Weber has specified that the selection of the grammatical constructs from a modeling grammar may depend on a person's perception and the way how each person is looking at that grammar [17]. Also he stated that the mapping between the grammatical constructs and the ontological concepts may differ from person to person. Therefore, even if some other researchers perform the same research study, there can be a possibility that they will get some different values than the ones specified in this paper.

ACKNOWLEDGMENT

We are indebted to Prof. Ron Weber (Faculty of Information Technology - Monash University, Australia) and Mr. Kapila Dias (University of Colombo School of Computing, Sri Lanka) for the helpful comments and reference suggestions given on this paper.

REFERENCES

[1] W. Wang and R. J. Brooks. (2007). Improving the Understanding of Conceptual Modelling. *Journal of Simulation*. [Online]. 1(3). pp. 153-158. Available: http://www.palgrave-journals.com/doifinder/10.1057/palgrave.jos.4250024

[2] S. K. Milton, J. Rajapakse, and R. Weber, "Conceptual Modeling in Practice: An Evidence-Based Process-Oriented Theory," in *Proc. 5th International Conf. on Information and Automation for Sustainability*, 2010, pp. 533-536.

[3] B. Anglim, S. Milton, J. Rajapakse, and R. Weber, "Current Trends and Future Directions in the Practice of High-Level Data Modeling: An Empirical Study," in *Proc.17th Europian Conf. of Information Systems*, 2009, pp. 1-13.

[4] J. Rajapakse and M. Orlowska, "Towards a Graphical Transactional Workflow Specification Language," in *Proc. Australian Systems Conf.*, 1995.

[5] J. Rajapakse, "On Conceptual Workflow Specification and Verification," M. S. Thesis, Dept. Comp. Science, The University of Queensland, Australia, 1996.

[6] J. C. Recker, M. Rosemann, M. Induska, and P. Green. (2009). Business Process Modeling: A Comparative Analysis. *Journal of the Association for Information Systems*. [Online]. 10(4). pp. 333-363. Available: http://eprints.qut.edu.au/20105/

[7] K. Siau. (2004). Informational and Computational Equivalence in Comparing Conceptual Modeling Methods. *Journal of Database Management*. [Online]. 15(1). pp. 73-86. Available: http://www.igi-global.com/article/informational-computational-equivalence-comparing-information/3306

[8] S. Hadjerrouit, "A Constructivist Approach to Object-Oriented Design and Programming," in *Proc.4th Annual SIGCSESIGCUE ITiCSE Conf. on Innovation and Technology in Computer Science Education*, vol. 31, no. 3, pp. 171–174, 1999.

[9] OMG, "The Unified Modeling Language Specification -- Version 1.4.2," *The Object management Group*, 2005.

[10] J. Rumbaugh, R. Jacobson, and G. Booch, *The Unified Modelling Language Reference Manual*, 2nd Ed. Addison-Wesley Professional, 2004, pp. 576.

[11] H. Gomaa, *Software Modeling and Design: UML, Use Cases, Patterns, and Software Architectures*, 1st Ed. Cambridge University Press, 2011, pp. 592.

[12] A. Dennis, B. H. Wixom, and D. P. Tegarden, *Systems Analysis and Design with UML Version 2.0: An Object-Oriented Approach*, 2nd Ed. John Wiley & Sons, 2005, p. 525.

[13] J. Evermann and Y. Wand. (January 2005). Ontology Based Object-Oriented Domain Modelling: Fundamental Concepts. *Requirements Engineering*. [Online]. 10(2). pp. 146-160. Available: http://www.springerlink.com/index/10.1007/s00766-004-0208-2

[14] J. Evermann and Y. Wand. (2009). Ontology Based Object-Oriented Domain Modeling - Representation Behaviour. *Journal of Database Management*. [Online]. 20(1). pp. 48–75. Available: http://www.igi-global.com/viewtitlesample.aspx?id=3400

[15] P. Tilakaratna and J. Rajapakse, "Forward Engineering the Object Oriented Analysis and Design," Presented at the 5th Malaysian Conference in Software Engineering, 2011, pp. 107-112.

[16] P. Tilakaratna and J. Rajapakse. (2012). Ontological Framework for Object-Oriented Analysis and Design. *American Journal of Engineering and Applied Sciences*. [Online].5(3). pp. 251-260. Available: http://thescipub.com/abstract/10.3844/ajeassap.2012.251.260

[17] R. Weber, "Ontological Foundations of Information Systems," *Coopers & Lybrand and the Accounting Association of Australia and New Zealand*, no. 4, pp. 212, 1997.

[18] J. Evermann and Y. Wand. (January 2005). Toward Formalizing Domain Modeling Semantics in Language Syntax. *IEEE Transactions on Software Engineering*. [Online]. 31(1). pp. 21-37. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1392718

[19] J. Kim, S. Park, and V. Sugumaran. (January 2008). DRAMA: A Framework for Domain Requirements Analysis and Modeling Architectures in Software Product Lines. *Journal of Systems and Software*. [Online]. 81(1). pp. 37-55. Available: http://linkinghub.elsevier.com/retrieve/pii/S016412120700088X

[20] Y. Wand and R. Weber. (October 1993). On the Ontological Expressiveness of Information Systems Analysis and Design Grammars. *Information Systems Journal*. [Online]. 3(4). pp. 217-237. Available: http://doi.wiley.com/10.1111/j.1365-2575.1993.tb00127.x

[21] P. Angeles, *Dictionary of Philosophy*, 2nd Ed. Barnes & Noble, 1981, pp. 335.

[22] M. A. Bunge, *Treatise on Basic Philosophy: Volume 3: Ontology I: The Furniture of the World*, D. Reidel Publishing Company, 1977.

[23] M. A. Bunge, *Treatise on Basic Philosophy: Volume 4: Ontology II: A World of Systems*, D. Reidel Publishing Company, 1979.

[24] P. Green, M. Rosemann, M. Indulska, and C. G. Manning. (2007). Candidate Interoperability Standards: An Ontological Overlap Analysis. Data & Knowledge Engineering. [Online]. 62(2). pp. 274-291. Available: http://linkinghub.elsevier.com/retrieve/pii/S0169023X06001613

[25] P. Green, M. Rosemann, and M. Indulska. (2005). Ontological Evaluation of Enterprise Systems Interoperability Using ebXML. *IEEE Transactions on Knowledge and Data Engineering*. [Online]. 17(5). pp. 713-725. Available: http://ieeexplore.ieee.org.ezproxy.lib.monash.edu.au/stamp/stamp.jsp?tp=&arnumber=1411749

[26] C. D. Keen and C. Lakos, "Analysis of the Design Constructs required in Process Modeling," in *Proc. of the International Conf. on Software Engineering*, pp. 434-441, 1996.

[27] P. Green and M. Rosemann, "Integrated Process Modelling. An Ontological Evaluation," *Information Systems*, pp. 73-87, 2000.

Prabodha Tilakaratna received the B.Sc. degree in Information and Communication Technology from the University of Colombo School of Computing, Sri Lanka in 2010, BCS (British Computer Society) diploma in 2009. Currently she is a PhD student at the Faculty of Information Technology, Monash University. Before starting the PhD, she has been involved with teaching and research in software engineering at the University of Colombo School of Computing, Sri Lanka.

Jayantha Rajapakse received the B.Sc. degree from the University of Peradeniya, Sri Lanka in 1981, Research Masters degree in Computer Science from the University of Queensland, Australia in 1997, and the Ph.D. degree in Information Systems from the University of Melbourne, Australia in 2006. Currently he is a Senior Lecturer at the Faculty of Information Technology, Monash University. He has been involved with teaching and research in information systems and computer science for over 15 years at the University of Queensland, the University of Melbourne, Queensland University of Technology, Manchester Metropolitan University (Sri Lanka), and Monash University.