A proposed Ontology to support Modeling Diagrams

Nisreen Innab[1], Nidal Yousef[2], and Mohammad Al-Fayoumi[3]

[1] Computer Information Systems, College of Business Administration
Jeddah ,KSA

[2]Information Systems, Isra University
Amman, Jordan

[3] Information Systems, King Abdul Aziz University
Jeddah ,KSA

Abstract
This paper investigates ontology. Ontology exhibits enormous potential in making software more efficient, adaptive, and intelligent. It is recognized as one of the areas which will bring the next breakthrough in software development. Ontology specifies a rich description of the terminology, concepts and properties explicitly defining concepts. Since understanding concepts and terms is one of the difficulties in modeling diagrams, this paper suggests an ontology aiming to identify some heavily used modelling diagrams concepts to make them easier to perform

Keywords: *Ontology, modeling, concept, requirement*

1. Introduction

In systems engineering and software engineering, requirements analysis encompasses all of the tasks that go into the instigation, scoping and definition of a new or altered system. Requirements analysis is an important part of the system design process, whereby requirements engineers and business analysts, along with systems engineers or software developers, identify the needs or requirements of a client. Once the client's requirements have been identified, the system designers are then in a position to design a solution **[1]**.
The requirement phase based on robust conceptual models. Ontologies are a promising means to achieve these conceptual models, since they can serve as basis for comprehensive information representation and communication.

During requirements stage many modelling can be used for the similar or different systems. It depends on the multiple views of the developers and the types of the systems. Authors of requirements use different terminology and hence the same term is applied to different concepts and different terms are used to denote the same entity. The same semantic of multi concepts may use in different modelling. For example the concept entity which is used in (entity) relationships modelling has a close meaning to the concept (class) which has been used in unified modelling language. The differences lead to difficulties of understanding concepts during modelling and confusion, and diminish exchange of knowledge - between developers and domain experts.

Therefore we propose to suggest a framework that aims to identify some important concepts that are heavily used at different modelling diagrams, to be easier to perform.
In this paper, we introduce a framework for building and using ontology for modelling diagrams in order to standardize their different concepts. We can support modelling by using ontology (meaning of things), which helps the software developer and the user of the system to define and understand the concepts and their definitions easily.

2. The Proposed ontology

We propose framework that aims to identify some important concepts that are heavily used at different modelling diagrams, to be easier to perform.The framework consist the following stages:We select some of well-known modelling diagrams of requirements process.
Each selected modelling diagrams has its semantic and standards (ontology) .We extract and refine the concepts of those selected modelling diagrams. Then we arrange each of the extracted concepts according to their frequency of existing in all the selected modelling diagrams by using specific program. We specify a collection of concepts that have the highest frequency use in the selected modelling diagrams

Then we emphasize the importance of the thirty highest frequencies concepts for our results in both approaches by using percentage technique, which depends on the proportion of how many concepts of the thirty highest frequency concepts is part of each modelling diagram concepts. We choose the first thirty highest frequency concepts from our results in each approach in order to limit our study where the frequency was increased steadily for the first thirty highest frequency concepts. We use Bunge–Wand–Weber (BWW) representation model for evaluating, comparing our proposed ontology. BWW is a well-known model that had been.

3. Reasons for using ontology with modelling

Modelling can be made easier by using ontology. The reason is that the ontology provides the basic concepts and their relations that is easy to grasp even for non-experts on modelling.

Although the concept of ontology has been around for a long time in philosophy, in recent years it has become known in the computer world as a machine-readable vocabulary that is specified with enough precision to allow differing terms to be precisely related.An ontology defines the common words and concepts (meanings) used to describe and represent an area of knowledge. Ontologies are used by people, databases, and applications that need to share domain information (a domain is just a specific subject or area of knowledge, like medicine, counterterrorism, imagery, automobile repair, etc.) Ontologies include computer-usable definitions of basic concepts in a domain and the relationships among them.

Ontologies encode knowledge in a domain and also knowledge that spans domains. In this way, they make that knowledge reusable **[2].** Ontology shows enormous potential in making software more efficient, adaptive, and intelligent. It is recognized as one of the areas which will bring the next breakthrough in software development. The idea of ontology has been welcomed by visionaries and early adopters for example, ontology has been used in medical informatics studies, and the community produced popular tools such as Protégé ontology editor. However, it has failed to appeal to the majority users of the mainstream, at least until recently. It is said that the idea was too mysterious for ordinary people to understand. There is no standard way to do things with ontology, but so many different proprietary ways. There were not enough tools for programming ontologies and managing various aspects of the life cycle of ontologies **[3]**.

We make use of ontology with modelling diagram in this paper because of its significant nowadays, its improvement than the past, because there several applications which use ontology, and the ontology specifies a rich description of the terminology, concepts relevant to a particular domain or area of interest. Ontologies might be important for Software Engineers because they have to share their used previously in systems analysis and design research.

conceptualizations with each other, and with their user communities.

There are a number of problems ontology solves particularly well, such as problems with information integration, model transformation, translation, data cleansing, search, navigation, text understanding, document preparation, speech understanding, "active" menus and forms, question-and-answer issues, etc. Developers should consider whether there are other innovative ways of using ontology for solving their problems. Once developers are comfortable with the value ontology brings in, they can go ahead and populate the ontology for the solution.

One of the main difficulties in modelling diagrams understands concepts and terms by the system developer and users without complexity and conflicts. To solve this problem this research proposes to build ontology and use it during modelling to reduce ambiguity of modelling diagram concepts.

4. The Framework for building the initial ontology .

The purpose of this paper is to suggest a framework that aims to identify some important concepts that are heavily used at different modelling diagrams, to be easier to perform. The framework consists of: selecting modelling diagrams, extracting concepts, found relation between each concept with all selected modelling diagrams.

Since there is a huge number of modelling diagrams concepts, we intend to minimize those concepts on a small number that are rigorously very important and have the majority used in most of modelling diagrams. Therefore, we use two approaches to emphasize the significance of the results and strictly build our ontology on sturdy base.

4.1 Selecting modelling diagrams

In the field of software engineering in particular in requirement phase, several and different modelling diagrams are used. We selected 56 diagrams and collected their definitions from multi recourses **[4, 5, 6, 7, 8, 9, 10, 11].** Snapshot of the results of this step were as in table (4,1)

4.2 Extracting Concepts

For each selected modelling diagram we extract most of their concepts based on many references that explain, define each modelling diagram, classify its concepts with

Diagram Name	Diagram Definition
Activity Diagram	Depicts high-level business processes, including data flow, or to model the logic of complex logic within a system.It shows the overall flow of control.
Use Case Diagram	The use-case diagram depicts a collection of use cases, actors, their associations, and optionally a system boundary box. When modelling requirements a use case diagram can be used to model the context of your system, indicating the major external entities that your system interacts with.
State Machine Diagram	A State Machine Diagram models the behaviour of a single object, specifying the sequence of events that an object goes through during its lifetime in response to events. A State Machine diagram illustrates how an element, often a class, can move between states classifying its behavior, according to transition triggers, constraining guards, and other aspects of state machine diagrams that depict and explain movement and behaviour

Table (4.1): Definition of Selected Modelling Techniques and Diagrams

their notation, the relationship between those concepts, their utilization and the modelling diagram itself **[4, 5, 8, 9, 12,13, 14,15, 16, 17, 18, 19, 20].** In the following, we show snapshot of the table that we construct from this step.

Diagram	Concepts
Activity Diagram	Activities, Actions, Action Constraints, Control Flow, Initial Node, Final Node, Objects and Object Flows, data store, Decision and Merge Nodes, Fork and Join Nodes, Expansion Region, Exception Handlers, Interruptible Activity Region, Partition.
Use Case Diagram	Actors, Use Cases, Name and Description, Requirements, Constraints, Scenarios, Scenario Diagrams, Extending Use Cases, Extension Points, System Boundary.
State Machine Diagram	States, Initial and Final States, Transitions, State Actions, Self-Transitions, Compound States, Entry Point, Trigger, Exit Point, Choice Pseudo-State, Junction Pseudo-State, Terminate Pseudo-State, History States, Concurrent Regions.

Table (4.2): Snapshot for Each Modelling Diagram with its Concepts

Then we define each concept that we arrange in categories as we show in the previous table without redundancy. Table (4.3) shows snapshot of the concepts with their descriptions that we extract from all the previous selected modelling diagrams.

Concept	Concept Definition
Abstract Class	A class that cannot be directly instantiated. A class which cannot instantiate objects; it must be inherited
Activity Final	The end point of an activity diagram
Classification	The assignment of an object to a classifier
Client	A classifier that requests a service from another classifier.
Entity	Person, place, thing. It is an object or concept about which you want to store information.

Table (4.3): The Extracted Concepts with their Definitions

4.3 Relation between each concept with all selected modelling diagrams

We found the relation between each concept with all selected modelling diagrams definitions by using access programme as a tool. Where we create two tables, the first one contains the modelling diagrams definitions and the second table contain all the concepts then, we create a query to find the number of times (frequency) for each concept in which it appears in all selected modelling diagrams definitions.

We show some of the results in table (4.4) which contains frequency column which indicates number of occurrence for each concept in all selected modelling diagrams definitions.

Concept	Frequency
Model	30
System	28
Diagram	25
Element	18
Action	15

Table (4.4): Concepts Relationship with Modelling Diagrams

We notice from the results that there is a different frequency for each concept. Some have a high frequency as: model, system, diagram, element, action, object, user, class, interaction, etc. That means the concepts with high frequencies are significant as they are related to many modelling diagrams. Thus we can define it once and use it in all the selected diagrams.

5. Ontological Analysis

We agree that ontological analysis and evaluation is one of several approaches that should be used to improve the modelling diagrams. We evaluate the results by percentage technique that depends on the frequency.

We choose the 30 highest frequencies of the results that shown in table(4.4) to limit our study where the frequency was increased steadily for the first thirty concepts.

To emphasize the importance of the thirty highest frequencies concepts for our results we use percentage technique as follows:

- Identify all the selected modelling diagrams.
- Classify the concepts for each modelling diagram in a category.
- Then count the concepts for each category
- Demonstrate the thirty highest frequency concepts, then find and remark if any of those concepts exist in each category.
- Count remarked concepts for each category.
- Define the proportion.

- Number of how many concept of the 30 highest frequency concepts exist in this category : Number of all concepts for each category.
- Exchange the values of proportion to 100% percentage.

Table (5.1) shows the results of the previous steps. For the first diagram Activity diagram we identify in the third column a collection of its important concepts as action, actor, and constraint. Then we set in the fourth column number of all its concepts that we identify in column three. We use the category of the highest frequency concepts that we outline in table (4.4), to search about those specific concepts if they exist or not in the collection of the Activity concepts that we show in column three. After that, we count how many important concepts exist and write the number in column five. The proportion between the numbers of Activity concepts (column four) to the number of specific existing concepts (column five) are 7:34, which present in column six. The proportion 7:34 indicates 20.6% which shows in the last column of the table.
Indeed, the other modelling diagrams can be recognized in the same way for Activity diagram.

According to the results of Table (5.1), the highest percentage for the frequency of the specific thirty concepts rate to the number of the concepts of each modelling diagram is 100% for domain model, dialogue model, application model, object role modelling, and joint application development. While the lowest percentage is for block diagram, function block diagram, and mind map. The table shows the moderate percentage (50%) is for composite structure diagram, component diagram, subsystem diagram, high-level petri nets, and timed petri nets.
A central idea behind our approach is to identify common significant modelling concepts by (1) breaking modelling constructs and model elements down into ontologically primitive concepts and (2) mapping each concept onto a common ontology

6. Comparison with BWW

Given the importance of using and the potential use of ontologies over the past years, the principal question then becomes: How do we compare and evaluate different ontologies for determining their strengths and weaknesses for the purpose required?
Only limited guidance is provided regarding the selection and evaluation of ontologies. We use Bung- Wand-Weber (BWW) model for evaluating, comparing our ontology. BWW is a well-known model that had been used previously in systems analysis and design research.

BWW is a representation model that defines ontological constructs for information systems. According to these

Diagram	Concepts (Con.)	No. All Con.	No. 30 Con.	Proportion	100 %
Activity Diagram	Action ,Actor, Constraint, Control Flow, Initial Node, Final Node, Object and Object Flow, Data Store, Division and Merge Nodes, Fork and Join Nodes, Expansion Region, Exception Handlers, Interruptible Activity Region, Partition, Association, Class, Association Class, Attribute, Base Class, Branch, State, Final State, Transition, Symbol, Control, Thread, Rendezvous, Swimlane ,Diagram, Activity, Data.	43	7	7:43	20.6
Use Case Diagram	Actor, Use Case, Name, Description, Requirement, Constraint, Scenario, Scenario Diagram, Extending, Extension Points, System Boundary, Visibility, Class, Property, Component, Core Element, Attribute , Operation , Type, Description, Association, Extend, Generalization, Include, Diagram, Element, System.	27	10	10:27	37.0

Table (5.1): Evaluate the results by Percentage Technique

constructs the completeness and efficiency of a modelling technique can be defined **[21].** It defines five fundamental constructs which are: things, property, state, transformation, and stable state. A thing is the existence of a thing in the world. Properties are relations that map a thing to some values. A stable state is a state in which a thing will remain unless forced to change by an external event. Mapping between one state to anther is called transformation.
The following Table (6.1) present comparison of BWW constructs with the specific concepts of our results**[22, 23, 24, 25, 26].**

Definition of BWW concepts	BWW	Our Ontol-ogy	Equiv-alent?
The elementary unit in the BWW ontological model. The real world is made up of things.	Thing	Object, Entity	Yes
Things possess properties. A property is modeled via a function that maps the thing into some value.	Property	Attribut e	Yes
A set of things that can be defined by their possessing a particular set of properties	Class	Class	Yes
TOTAL Concepts	30	30	25

Table (6.1): Comparison with BWW Constructs

The results of the comparison between the most known concepts of BWW representation model (30 concepts) and the most important concepts of our result of our initial ontology (30 concepts) shows that twenty five concepts represent 83.33% have the same semantic.

One of the contributions of this comparison is to provide a clear ontological definition of modelling diagrams constructs. Our analysis shows that the ontology we are constructing has some ontological strength as the results identify the common core constructs of modelling diagrams (for example, object, attribute, class). The analysis and evaluation shows that many of our ontology construct are well matched with the BWW-model, but also we suggest several concrete improvements in future to our ontology.

7. Conclusions

The purpose of this paper is to enhance the existing modelling diagrams which are used during requirement phase by building ontology which includes defining terms and classifying them. In order to reduce ambiguity of modelling diagram concepts, and to understand concepts and terms by the system developer and users without complexity and conflicts.

Many modelling can be implemented in various ways. There are different concepts that the developers and the customers must learn. The difference of modelling in the real world splits the developers in different schools; each one may focus on different aspects of the world and show that developers understand the world differently. Ontology can enhances the existing modelling techniques which are used during requirement phase. In addition, ontology includes definitions, classifications and formalization of terms.

This paper concentrates on modelling in requirement phase. Authors of requirements use different terminology and hence the same term is applied to different concepts and different terms are used to denote the same entity. Therefore it leads to difficulties in the understanding of the concepts and terms by the system developer and users without complexity and conflicts. One way that provides support to solve this problem is to build ontology and use it during modelling.

We argue the constructs of modelling diagrams have to be communicated using some commonly understood "concepts."

Future Work

In our future research, we recommend to apply the framework of the proposed ontology with other software development process. And apply these ideas with the use of an Arabic ontology.

8.References

[1] Barry Boehm, "A view of 20th and 21st century software engineering", Proceedings of the 28th international conference on Software engineering, Shanghai, 2006, ISBN:1-59593-375-1, ACM, pp: 12 – 29

[2] Leo Obrst, "Ontologies for Semantically Interoperable Systems", CIKM'03, 2003.

[3] Milton, S.K., & Kazmierczak, E., "Enriching the Ontological Foundations of Modelling in Information Systems", Proceedings of Workshop on Information Systems Foundations: Ontology, Semiotics, and Practice, (Ed, Dampney, C.), 2000, pp: 55-65.

[4] Stephen J. Mellor, "A Comparison of the Booch Method and Shlaer-Mellor OOA/RD", by Project Technology, Inc, 1993.

[5] Recker, Jan C. and Indulska, Marta, "An Ontology-Based Evaluation of Process Modeling with Petri Nets", IBIS - International Journal of Interoperability in Business Information Systems Vol.2 No.1, 2007:pp. 45-64.

[6] Malcolm Shroff & Robert B. France, "Towards a Formalization of UML Class Structures in Z", Computer Software and Applications Conference, 1997. COMPSAC '97. Proceedings., pp:646 – 651.

[7] Johannes Koskinen , Jari Peltonen , Petri Selonen , Tarja Systa , Kai Koskimies, "Model processing tools in UML", 23rd International Conference on Software Engineering, 2001pp:819-820.

[8]Yeol Song and Kristin Froehlich, "Entity-Relationship Modeling: A Practical How-to Guide," IEEE Potentials, Vol. 13, No. 5, 1994-1995, pp. 29-34.

[9] Kevin L.Mills, Hassan Gomaa, "Knowledge-Based Automation of a Design Method of Current Systems", IEEE Transaction of Software Engineering, Vol.28. No.3, 2003.

[10] Federico Vazques, "Identification of complete data flow diagrams ", ACM SIGSOFT Software Engineering Notes, Vol. 19 , Issue 3, 1994, pp: 36 - 40 .

[11] luciano baresi, mauro pezz`e, "Formal Interpreters for Diagram Notations", ACM Transactions on Software

Engineering and Methodology, Vol. 14, No. 1, 2005, Pages 42–84.

[12] Boris Litvak, "Behavioral Consistency Validation of UML diagrams", Proceedings of the First International Conference on Software Engineering and Formal Methods 2003.

[13] R. Juric and Il-Yeol Song, "The Assessment of Object-oriented Modleing Elements of the UML 1.1" 3rd Biennial World Conference. on Integrated Design and Process Technology, 1998.

[14] UML Semantics Appendix M1- UML Glossary version 1.0, 13 January 1997

[15] David N. Jansen, Holger Hermanns, "QoS modelling and analysis with UML-statecharts: the StoCharts approach", ACM SIGMETRICS Performance Evaluation Review, Vol. 32, Issue 4 2005 pp. 28 – 33.

[16] Jose Galindo, Angelica Urrutia, Ramon A. Carrasco, and Mario Piattini, "Relaxing Constraints in Enhanced Entity-Relationship Models Using Fuzzy Quantifiers", IEEE Transactions On Fuzzy Systems, vol. 12, no. 6, 2004.

[17] Veda C. Storey, "Comparing Relationships in Conceptual Modeling: Mapping to Semantic Classifications", IEEE Transactions On Knowledge And Data Engineering, vol. 17, no. 11, 2005.

[18] Swathi C . Solsi, Edward L. Jones, "Simple Yet Complete Heuristics for Transforming Data Flow iagrams into Booch Style Diagrams", Ada Letters, Volume XI, Number 2, 1991.

[19] Kevin L. Mills, Hassan Gomaa, "A Knowledge-Based Method for Inferring Semantic Concepts from Visual Models of System Behavior", ACM Transactions on Software Engineering and Methodology, Vol. 9, No. 3, 2000, pp. 306–337.

[20] Harald Storrle, "Structured nodes in UML 2.0 activities", Nordic Journal of Computing, Vol.11 , Issue 3 2004 pp. 279 – 302.

[21] Ahmad Kayed, Robert M.Colomb, "Using BWW Model to Evaluate Building Ontologies in CGs Formalism", Vol. 30, No.5, 2005,pp 379-398

[22] Michael Rosemann1, Jan Recker, Marta Indulska, Peter Green, "How have Process Modeling Grammars Developed over Time?", Proceedings of the 18th Conference on Advanced Information Systems Engineering, 2006, pp. 447-464.

[23] Michael Rosemann, Jan Recker, Marta Indulska, Peter Green, "How have Process Modeling Grammars Developed over Time?", Proceedings of the 18th Conference on Advanced Information Systems Engineering 2006, pp. 447-464.

[24] Michael Rosemann, Peter Green. "Integrating Multi-Perspective Views Into Ontological Analysis", ICIS 2000 pp. 618-627.

[25] Gretchen Irwin, Daniel Turk, "An Ontological Analysis of Use Case Modeling Grammar", Journal of the Association for Information Systems Vol. 6 No.1, 2005.pp.1-36.

[26] Andreas L. Opdahl, Brian Henderson-Sellers, "Ontological Evaluation of the UML Using the Bunge–Wand–Weber Model", Softw Syst Model, 2002 pp. 43–67.

Nisreen Innab received her B.S. degree in 1997. In 2003,she earned her High Diploma Degree in Computer Information Systems, in 2004 she earned Master Degree in Computer Information Systems. A Ph.D. was received in 2008 in Computer Information System, she is employed as full time lecturer, Assistant Professor and MIS department Chairperson at College of Buisness Administration (CBA) in Saudi Arabia, Jeddah from Oct. 2007 till now. She published three papers in International Conferences. Her current research interests are: data mining, machine learning, modeling and simulation, ontology, modeling diagrams.

Nidal Yousef received his B.S. degree in 1997. In 2001, he earned his Master degree in Computer Information System. A Ph.D. was received in 2008 in Computer Information System. He joined King Abdulaziz University, in KSA in 2009, as an Assistant Professor and in 2010 he moved to Al- Esra University in Jordan as Assistant Professor in the college of Computing and Information Technology. He has published 4 research papers in international journals and conferences.

Mohammad Al-Fayoumi received his B.S. degree in 1974. In 1977, he earned his Master degree in mathematics, and a postgraduate diploma in Computer Science was received in 1979. A Ph.D. was received in 1982 in the area of Computing Science. The last two degrees were awarded from Bucharest University; he joined the Yarmouk University, 1982 in Jordan, as an Assistant Professor and a Head of Computer Science Department. In 1986 he moved to Collage of Business Studies in Kuwait and then moved back to Jordan in Applied Science University as Associate Professor and a Head of Computer Science Department. In 2005 he moved to the Middle East University for Graduate Studies in Jordan as an Associate Professor and a Dean of Information Technology Faculty, he promoted to a professor rank in August, 2008. Currently, he is a professor and advisor of graduate studies at King Abdulaziz University, Saudi Arabia. His research interests include areas of information security, computer simulation, systems development, e-commerce, e-learning and internet security and algorithm analyzes and design. He has supervised many PhD's and master's degrees research thesis and projects of diverse areas. He has published more than 50 research papers in a multitude of international journals and conferences, in addition to a nine books in the area of computer sciences