

Masters of Computer and Information Sciences

ASSIGNMENT COVER SHEET - INDIVIDUAL	
Student Name and ID: LI, Mao Chuan; 14854389 Date: 2015/03/13	
Paper Name and Code: Software Requirements Engineering, 409220	
Assignment Name: Research Paper Analysis	Number of Words/Pages: 1600/6

In order to ensure fair and honest assessment results for all students, it is a requirement that the work that you hand in for assessments is your own work.

Please read and place an "X" in the boxes below to confirm that the statements are true of the work you are submitting.

Where I have used someone else's words, I have indicated this (using quotation marks and adding the reference). Where I have used other people's ideas or writing, I have indicated this by putting them into my own words and adding the reference.

X

Other than this, this assignment:

IS NOT copied from another student or from a previous assignment.

X

IS NOT copied from books, journals, Web pages or other sources.

X

HAS NOT been handed in by me or anyone else in any other course.

X

IS WRITTEN BY ME as the sole author

X

Complete this section only if the assessment is being submitted after the original assessment deadline

Extension granted? YES / NO

If an extension was granted, by whom:(name),

and until when: (date & time)

By filling out the form and submitting it electronically I confirm that it is a true and accurate statement about me, the named student.

Analysis Report for Paper *How UML is used*

Prepared for:	Jim Buchan
Prepared by:	Mao Chuan Li
Student ID:	14854389
Submit Date:	2015/03/06
Paper Name:	Software Requirements Engineering
Paper Number:	409220
Assignment No:	1a
Assignment Name:	Research Paper Analysis

1. Introduction

Since the proposal of the Uniformed Modelling Language (UML) to the Object Management Group (OMG) in 1996, UML has developed to 2.5beta version with more than 20 diagrams to visually describe the models of a system in design and implementation. Although UML has been “widely accepted as a modelling standard for OO software development” (Dobing & Parsons, 2006), and “a large number of practitioner articles and books and some contributions by academic researchers, have been devoted to articulating various aspects of the language” (Dobing & Parsons, 2006), there were rare researches about how exactly UML is used in real world businesses and how successful it helps in software requirements engineering.

In May 2003, Professor Dobing and Profession Parsons conducted a critical and influential research on above questions. They created a web survey open to all UML practitioners; collected 182 meaningful responses, which lasted a whole year. This report is to critically review the paper *How UML is used written* by the two professors published on magazine Communications of the ACM in May 2006.

2. Discussion

2.1. UML diagrams

In the paper, there are nine diagrams mentioned. Here are a list of brief descriptions of these diagrams:

2.1.1. Structural Diagrams

- Class Diagram - is the most fundamental and important diagram in the Unified Modelling Language (UML). It is composed of a collection of relevant classes, their attributes, operations, relationships and constraints. All classes defined in this model could be directly mapped to the Object-oriented programming classes.
- Object Diagram - is a representation of a collection of instances of classes at a specific time point, which could be thought of as a snapshot of the system.

- Component Diagram - is an overview of system components which can interact with each other by flexible interfaces. Generally all components can be replaced without affecting any other components.
- Deployment Diagram - is to render the topology of the physical components of target system. It contains all the hardware nodes and their relationships.

2.1.2 Behaviour Diagrams

- Use Case Diagram - is normally the first diagram to identify all the features needed to create for a project and depict the interactions between all kinds of users and the system and demonstrate what functionalities the system should provide. Once the use case diagrams created, corresponding detailed Use Case Narratives and User Scenarios could follow.
- Activity Diagram - is a high level description of system activities, typically derived from a use case. It gives an overview of system activities that happen in sequence, or parallel.
- Sequence Diagram - is a representation of interactions of different components/objects defined in the system in a time sequence. The core concept here is the messages exchanged between each component, so it is also called event diagrams sometimes.
- Collaboration Diagram - is a both static and dynamic diagram similar to sequence diagram to describe the interactions between objects. The distinguishing feature of it is that it can show the association of one object to another.
- Statechart Diagram - is later renamed to State Machine Diagram in UML2.0. This diagram is to represent a system or object with multiple states defined during its lifetime which could be triggered by an event.

2.2. Benefits and drawbacks of UML in requirements engineering

2.2.1. Benefits

The paper enumerated four types of benefits when using UML, all of them have been confirmed by the respondents' responses.

- a) Client Verification - "87% of respondents rated Use Case Narratives as useful for verifying and validating requirements with client representatives on the project team" (Dobing & Parsons, 2006).
- b) Clarifying Tech Understanding - "the Class and Sequence Diagrams were most useful for clarifying understanding of application among technical members of the project team" (Dobing & Parsons, 2006).
- c) Programmer Specifications - the UML Diagrams were also helpful for describing the programmer specification, with 89% of respondents rated Class Diagrams and 84% for Sequence Diagrams.
- d) Maintenance Documentation - the obvious and straight benefit of UML is that it can help effectively maintain and document the system design.

2.2.2. Drawbacks

Six potential drawbacks were listed for respondents to choose, and it turned out that only four of them are major problems when using UML:

- a) Not well understood by analysts - Both Class diagram and Activity diagram were reckoned as difficult to understand.
- b) Not useful for most projects - Statechart diagram were not fitting in most of the real projects.
- c) Insufficient value to justify cost - for Sequence and Use Case diagrams and Narratives, users did not think their outcome could justify the cost of using these diagrams.
- d) Information captured would be redundant - since Sequence and Collaboration are two isomorphic diagrams, providing similar information, the information

shown on Collaboration diagram may already have been shown on Sequence diagram.

As the paper concluded, all these drawbacks could be rooted to the shortage of experience and complexity of the UML language.

2.3. Conclusion of the paper *How UML is used*

In the end of the paper, the two professors summarized six conclusions based on this research:

- a) The Class, Sequence and Use Case Diagrams are mostly frequently used, while Collaboration Diagram are used least. 73% of respondents said Class diagram were used in two-thirds or more of the projects; 44% for Use Case Narratives; nearly 50% for Sequence diagram. Only 25% of respondents said Collaboration diagram were used in two-thirds or more of their projects.
- b) At least half of UML projects are not Use Case driven as expected before in UML literature. Based on the usage and non-usage frequency of each diagram, apparently, Use Case diagram is not always the first diagram created to drive the development.
- c) "Contrary to claims in the popular literature" (Dobing & Parsons, 2006), clients are more involved in the Sequence, Class, Collaboration and Statechart diagrams than expected. More than 30% of respondents reported that clients involved in these diagrams development; about half of respondents reported the involvement of review.
- d) The assumed "potential communication disconnect" concern does not actually exist in practice based on the survey data, since system analysts and programmers do not rely on Class and Sequence diagrams only, but rely on the Use Case Narratives as well to better understand the system requirements.
- e) Use Case Narratives is not the only diagram to capture all requirements of clients. Class, Statechart and Sequence diagrams have all been rated around 80% useful for providing new information for the projects.

- f) The UML language is too complex for users. From the "Table 2. Reason for not using some UML components" (Dobing & Parsons, 2006), it is easy to tell the UML diagrams are not well understood by all analysts. On the other hand, "generally, those with more UML experience made more use of UML components and for more purposes than those with less experience" (Dobing & Parsons, 2006). Obviously these both lead to the complexity of UML.

2.4. Research methods used

There are three research methods utilized in this paper: literature review, interview and web survey. The authors did not explain how they conducted the first two methods, but they should have played an important role in designing the survey questions.

To date, the survey is still open for all public, evolving with new questions. The paper is based on the data collected between March 2003 and March 2004. During that period, there were 182 usable responses submitted. Although the survey has been deliberately designed, the response number seems too few to make a reliable conclusion, which may be due to the long list of questions and many open questions in the forms.

Besides that, the scope of the survey were mainly focused on those practitioners related to OMG organization, who defined the UML. This may limit the research result only to those analysts having strong background of UML.

2.5. Credibility of the paper

The paper was written by Brian Dobing, an associate professor and Jeff Parsons, a University Research Professor, whose interests focus on object-oriented systems and reuse. The paper was published in the *Journal of the ACM*, a peer-reviewed scientific journal issued by the Association for Computing Machinery (ACM). It has 314 and 78 citations in Google Scholar database and ACM digital library, respectively by the time of March 13, 2015. Taking the above evidences into consideration, the credibility of the paper is reliable and trustworthy.

2.6. Influence from book *Agile Modelling: Effective Practices for Extreme Programming and Unified Process*

In the chapter 15 of this book, Ambler (2002) clearly expressed that "UML is too complex" (p. 171) and suggested to "Adopt a critical subset of the notation. If you only need 20 percent of the UML notation to do 80 percent of your modelling work" (p. 174). The paper's research result verified the book's opinion and agreed with his suggestion.

2.7. Influence on paper *Defining information systems as work systems: implications for the IS field*

The paper *HOW UML is used* did not give positive influence on the paper, in contrast, the paper questioned the validity of its conclusion, because there was another similar paper "give inconsistent results about how UML is used and the extent to which it is used by IT professionals"(Alter, 2008).

3. Conclusion

This report analysed in depth the paper *How UML is used*, found out the benefits and drawbacks of using UML in requirements engineering, and summarised the conclusions found in the paper.

The survey research method used in the paper is critically reviewed so that some limitations of that have been identified; despite of that, the credibility of the paper still remains reliable when considering the journal's publisher, the authors' academic attainments and the references of the paper in other articles.

In the end, two papers related to it has been reviewed to see how they influenced each other.

4. Reference

- Dobing, B., & Parsons, J. (2006). How the UML is used. *Communications of the ACM* 49(5), 109–113. doi: 10.1145/1125944.1125949
- Fowler, M. (2004). *UML distilled: a brief guide to the standard object modeling language*. Boston, MA: Addison-Wesley
- Ambler, S. (2002). *Agile Modelling: Effective Practices for Extreme Programming and Unified Process*. New York, NY: John Wiley
- Alter, S. (2008, October). Defining information systems as work systems: implications for the IS field. *European Journal of Information Systems*, 17(5), 448-469. doi:10.1057/ejis.2008.37
- University of Lethbridge. (2012, October 31). *Research Interests*. Retrieved on March 12, 2015, from <http://people.uleth.ca/~brian.dobing/research.htm>
- Memorial University of Newfoundland. (n.d.). *Jeffrey Parsons*. Retrieved on March 12, 2015, from <http://www.business.mun.ca/why-us/meet-our-people/faculty-instructor-profiles/jeffrey-parsons.php>