8-7-2011

Situations That Affect Modelers' Cognitive Difficulties: An Empirical Assessment

Palash Bera
Texas A&M International University, palash.bera@tamiu.edu

Situations That Affect Modelers' Cognitive Difficulties: An Empirical Assessment

Palash Bera
Texas A&M International University
palash.bera@tamiu.edu

ABSTRACT

Modeling guidelines are intended to help modelers develop conceptual models. Therefore it is assumed that these guidelines assist modelers in developing models. However research suggests that modeling guidelines are often complex. Thus there is a concern that if modeling guidelines are not well designed, modelers may not find them useful and will have difficulty in applying them. In the context of developing conceptual models, the effectiveness of ontological modeling guidelines has previously been tested. However, the usefulness of these guidelines has not previously been investigated. To test the usefulness of ontological modeling guidelines, four modeling situations were developed in UML class diagrams. Using a protocol analysis study, the cognitive difficulties of modelers placed into these situations were identified. The results indicate that the proper application and use of ontological modeling guidelines can significantly reduce the cognitive efforts of modelers. However, if the ontological modeling guidelines are inadequate, they increase rather than decrease modelers' cognitive efforts.

Keywords

Conceptual modeling, Modeling guidelines, UML, Protocol Analysis

INTRODUCTION

Conceptual modeling is a process by which domain experts and analysts create a representation of a domain (Weber, 2003). The models developed in this process, referred to as conceptual models, perform multiple critical roles during the definition, analysis, and communication of the requirements for the information system (IS). Prior research has demonstrated that conceptual modeling languages (such as UML) are complex in nature and have problems with learnability and ease of use (e.g. Siau and Loo, 2006). Therefore, it is not surprising that the process of developing conceptual models is also complicated (Soffer and Hadar, 2007).

Modeling support, in the form of modeling guidelines, intended to aid the modeling process has previously been proposed and tested (e.g. Bera, Burton-Jones and Wand, 2011; Evermann and Wand, 2006). However, there is a lack of research on how these guidelines, which provide modeling support, should be conveyed to the modelers (Soffer and Hadar, 2007). In particular, there is a concern that if the modeling guidelines are not well designed, then the modelers will not find them useful and will have difficulty in understanding and applying them. This might compromise the purpose of using the guidelines, which leads to the question *"how useful are the modeling guidelines in developing conceptual models?"* Accordingly, the objectives of this paper are to *identify situations when use of modeling guidelines might adversely affect modelers' cognitive difficulties* and to *analyze the extent of cognitive difficulties of modelers in such situations.*

To limit the scope of this paper, the modeling guidelines are restricted to developing UML association classes and guidelines that are derived from ontological theories are used. The next section describes conceptual modeling and the role of modeling guidelines in conceptual model development. Following this section, several modeling scenarios for developing UML association classes are presented. The empirical testing of these scenarios is then described, followed by the conclusion.

CONCEPTUAL MODEL DEVELOPMENT AND THE ROLE OF MODELING GUIDELINES

Conceptual modeling

Some popular conceptual models used in practice are UML class diagrams and Entity Relationship (ER) diagrams (Davies et al., 2006). An example of a conceptual model describing a patient registration situation, modeled in a UML class diagram is presented in Figure 1. The figure shows the concepts and relationships among Patient, Hospital, and Registration.

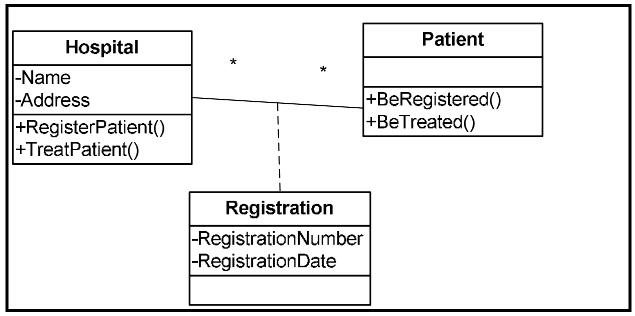


Figure 1: Example of a conceptual model in UML class diagram

UML has been widely used for IS design and software modeling (Davies, et al., 2006; Fettke, 2009) and it has also been suggested for use as a conceptual modeling language. A key advantage of using UML as a conceptual modeling language is it can be used both as an analysis language and as a design language using the same set of notations (Evermann and Wand, 2005).

Modeling guidelines

To develop a conceptual model such as Figure 1, it is important that the UML constructs are precisely defined. However, UML constructs such as object, class, and operation have not been precisely defined within the context of modeling application domains (Evermann and Wand, 2005). This might create problems for modelers when they are modeling business domains using these language constructs. To alleviate this problem, ontology based frameworks can be used to develop modeling guidelines. Ontology-based frameworks can provide clear modeling guidelines, which can be applied to conceptual modeling. Several frameworks exist that are based on different types of ontological foundations. For example, Evermann & Wand (2005) used Bunge's ontology to suggest rules for developing conceptual models in UML. Similarly, to develop conceptual models in UML, Guizzardi et al. (2004) suggested ontological rules that are based on an upper level ontology. The usefulness of these sets of rules has been empirically tested in several studies (e.g. Evermann and Wand, 2006; Soffer and Hadar, 2007).

It is possible that different modelers may come up with different conceptual models even if they are given the same domain and modeling technique. This phenomenon is referred to as "model variation" and it is quite common in conceptual modeling (Hadar and Soffer 2007). Model variations can be attributed to each modeler's expertise and training. To reduce these variations, Hadar and Soffer (2007) tested several modeling rules for developing UML class diagrams. They conclude that providing the modelers with a set of technical rules is not enough for achieving the desired change in the modeling process. They found that modelers experience difficulties when applying the rules, especially if a large set of rules is provided to the modelers.

The previous paragraph indicates that in spite of availability of modeling guidelines, it is important to unambiguously present the guidelines to the modelers. Although, earlier research has tested the effectiveness of the modeling guidelines in the context of developing conceptual models (e.g. Hadar and Soffer, 2007), the usefulness of these guidelines has not been investigated. In the next section we test such usefulness by developing situations involving modeling guidelines and describe the possible effects of these situations on modelers' cognitive difficulties.

SITUATIONS FOR MODELING ASSOCIATION CLASSES

To develop situations, we focused on a set of ontological guidelines proposed by Evermann and Wand (2005) for developing UML association classes. The concept of association was chosen, as it is quite popular in object oriented domain (Milicev, 2007).

Modeling association class in UML

In UML, association is defined as the relationship between the instances of classes and an association class is defined as an association that has attributes or operations of its own (Hoffer et al., 2007). Based on Bunge's ontology (1977), Evermann and Wand (2005) proposed a set of guidelines (Table 1) that a modeler can use to develop association classes in UML.

Table 1: Ontological guidelines to model UML classes and association classes

1)	Substantial entities should be identified and modeled as classes
2)	A class must have an attribute, a method, or both
3)	Properties of the substantial entities must be modeled as attributes and not classes
4)	Mutual properties must be represented as attributes of association classes

The essence of these guidelines is mutual properties should be represented as attributes of the association classes. In Bunge's ontology (1977), *mutual* properties exist between two or more *things,* where things denote objects that might exist in the real world. For example, salary is a mutual property between a person and an organization. To model association classes, several other guidelines should not be violated (Table A1, Appendix A). For brevity, the details of how these guidelines were derived are not discussed in this paper.

To apply the above mentioned guidelines on a domain description, a modeler will have to identify the mutual (or shared) properties of classes from the description. For example, if guidelines are applied to a simple domain description- "*a patient is registered in a hospital*" then Hospital and Patient can be identified as classes and the mutual properties of these classes RegistrationNumber and RegistrationDate should be modeled as attributes of the association class termed Registration. This will result in Figure 1. Typically, mutual properties are not mentioned in a domain description and modelers have a choice of using or not using the guidelines (Table 1). Accordingly, four situations (Table 2) arise that a modeler can face while modeling association classes.

Table 2: Situations faced by modelers in modeling association classes

Situations
1. A modeler uses the guidelines and applies them to a domain description along with the mutual properties
2. A modeler uses the guidelines and applies them to a domain description without the mutual properties
3. A modeler does not use the guidelines and develops models using a domain description along with the mutual properties
4. A modeler does not use the guidelines and develops models using a domain description without the mutual properties

The application of guidelines involves (1) identifying the attributes from the domain description, in particular mutual properties and then (2) relating these properties to the association class. Therefore if the domain description comes along with the list of mutual properties, then it can be considered as complete, because both domain description and mutual properties are required to apply the guideline. Accordingly, if only the domain description is provided to a modeler, we refer it to as an *incomplete* domain description and if the domain description and mutual properties are provided together, then we refer to it as a *complete* domain description. Thus in the four situations described in Table 2, domain descriptions can be manipulated in terms of *completeness* and modeling guidelines can be manipulated in terms of *availability* to the modelers. Now we present examples of models created if modelers are placed in these four modeling situations (Figure 2).

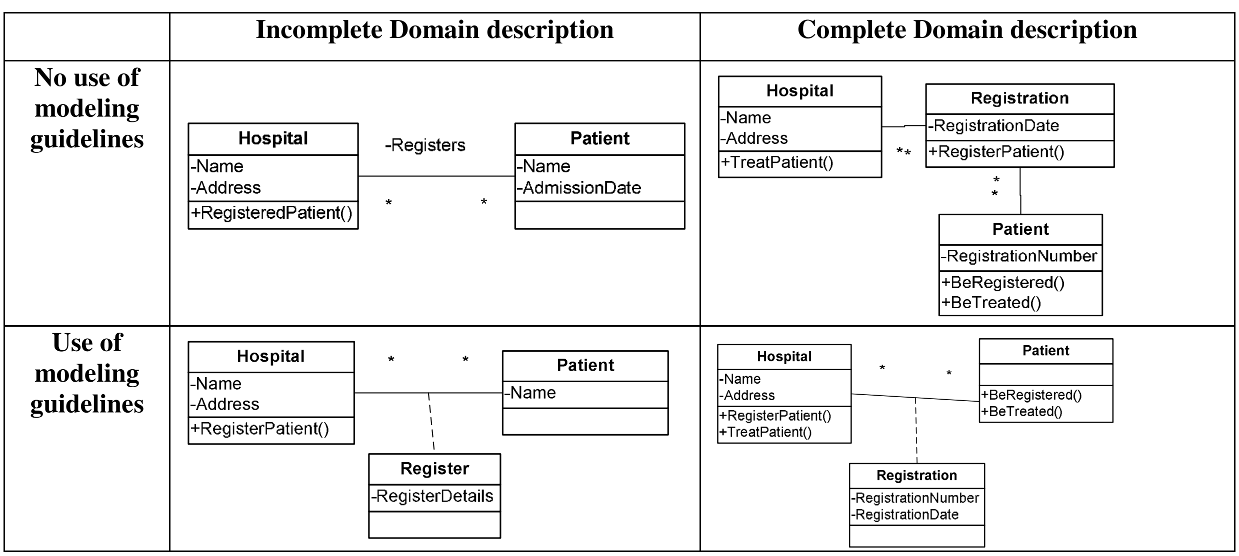


Figure 2: Example of models created in four situations

Application of the guidelines, along with the complete domain description, results in a detailed model such as Figure 1 (also shown in the lower right corner of Figure 2). The models that can be created by modelers in other three situations described in Table 2 are shown in the other areas in Figure 2. Note, that it is unlikely that modelers who do not use guidelines will create association classes (top two quadrants of Figure 2).

Cognitive difficulties in modeling association classes

Steps during conceptual model development such as modeling association classes involve cognitive difficulties of modelers, as it is a complex task. Cognitive difficulty arises due to the fact that humans have limited cognitive processing capacity to perform a complex task (Sweller, 1988). Table 3 mentions the possible extent of cognitive difficulties (high or low) faced by modelers in the four situations (also referred to as quadrants) previously mentioned.

Table 3: The extent of cognitive difficulties that modelers may face

	Incomplete Domain description	**Complete Domain description**
No use of modeling guidelines	Low cognitive difficulty (Quadrant 1)	High cognitive difficulty (Quadrant 2)
Use of modeling guidelines	High cognitive difficulty (Quadrant 3)	Low cognitive difficulty (Quadrant 4)

The extent of cognitive difficulties of modelers in quadrants 1 and 4 can be predicted to be low for different reasons. Modelers in quadrant 1 will not get any guidance in developing conceptual models, therefore they will develop models based on their prior modeling experience. Unless these modelers are given modeling scenarios which are very unfamiliar to them, it is expected that they will have few cognitive difficulties while developing the conceptual models. Modelers in quadrant 4 will get more information to develop model (in terms of domain description and guidelines) than modelers in quadrant 1, but as there is a perfect match between modeling guidelines and complete domain description, it is expected that they will have few cognitive difficulties.

It is expected that modelers in quadrants 2 and 3 will face many cognitive difficulties, as these modelers are given incomplete information to develop models. Modelers in quadrant 2 are given complete domain information but in the absence of guidelines, they might be unsure as to how to use the additional information (i.e. list of mutual properties) provided to them. On the other hand, modelers in quadrant 3 are given guidelines but are not provided with the complete domain information which is required for applying the guidelines. Thus it is expected that modelers in quadrants 2 and 3 will have more cognitive difficulties than those in quadrants 1 and 4.

Our interests is in finding out the extent of the cognitive difficulties of modelers who are placed in quadrants 2 and 3 situations, as these quadrants represent situations where modelers might have significant cognitive difficulties in developing conceptual models. Further, we intend to compare the extent of cognitive difficulties of these modelers with those placed in quadrants 1 and 4.

To compare the cognitive difficulties of the four groups we used the notion of *breakdowns*. A breakdown can be defined as the difficulty that a modeler faces when he/she tries to develop model from a domain description. Problem solving theory (Newell and Simon, 1972) suggests that when breakdowns occur during problem solving, subjects may either cycle back to the problem representation stage, or if they cannot overcome the difficulty, they might simple give up on the problem. The number of breakdowns serves as an indication of the cognitive difficulty of performing a task (Bera, Krasnoperova, and Wand, 2010). Breakdowns can be classified into two types: *explicit* and *implicit*. Explicit breakdowns occur when subjects specifically verbalize the difficulty that they faced in modeling. Implicit difficulties occur when modelers start a modeling activity (such as creating a class) and instead of completing it, they switch to another activity (such as creating another class). A plausible reason for this switch is modelers lack the knowledge on how to complete that activity and thus were forced to abandon it. Based on the above discussion, we identified situations that might bring different levels of cognitive difficulties to modelers. Accordingly we propose:

P1: Modelers situated in quadrants 1 and 4 will have fewer breakdowns when developing the UML class diagrams in comparison to the modelers situated in quadrants 2 and 3.

EMPIRICAL STUDY

One technique to identify the cognitive difficulties of the modelers is to ask them about the difficulties that they faced after the modeling task is complete (e.g. using questions related to perceived usefulness of the guidelines). However, use of such method might be inappropriate as modelers might not be able to express clearly the difficulties they had faced. Therefore in this study, to compare the cognitive difficulties of the modelers placed in the four quadrants and thus test the proposition, a verbal protocol analysis study was conducted. We followed the method suggested by Vessey and Conger (1994) who examined difficulties, or breakdowns experienced by subjects using graphic models to solve problems. Verbal protocol analysis method was used since the data obtained using this method reveals the mental processes that take place as individuals work on problem solving tasks (Ericsson and Simon, 1984; Newell and Simon, 1972). To use this technique, subjects are required to verbalize their thought processes and strategies as they create the UML class diagrams. By using a verbal protocol analysis study, the difficulties faced by the modelers can be directly identified and quantified. Next we describe the design, subjects, tasks, procedures, and results of the protocol analysis study.

Design and Subjects

A between-subject study was designed where each subject was placed in one of the four quadrants (Table 3). Subjects were 20 graduate students (5 placed in each quadrant) who were enrolled in an IS design course and were provided course credit for participation. Use of students as subjects is a common practice in conceptual modeling empirical studies (Burton Jones & Meso, 2006). Prior to the study, all subjects were trained extensively in developing UML class diagrams and they had taken two courses (IS Analysis and Database Management) that involved database modeling. To ensure that subjects were adequately trained in developing UML class diagrams, three UML class development assigments were conducted prior to the study. All subjects performed well on these assignments.

Tasks and procedures

All subjects were provided with a list of UML class diagram concepts and their definitions (Table B1). Subjects were placed randomly in one of the four groups. Subject placed in groups where no guidelines were used (quadrants 1 and 2) received general modeling rules to develop UML class diagrams (Table B2). These rules were developed from standard modeling practices and did not refer to any ontological concepts. Subjects in the other two groups (quadrants 3 and 4) received three sets of materials as ontological guidelines. First, they received a short description of the two ontological concepts *substantial entity* and *mutual property* (Table B3). These concepts were provided as they are crucial to understanding the ontological guidelines. Second, the ontological guidelines (Table 1) were placed into Table B2 and were provided to the subjects. Third, the rest of the guidelines were provided (Table A1) to these subjects. Then all subjects practiced developing UML class diagrams using the materials provided to them. Finally, subjects were given a domain description with or without a set of mutual properties, depending on which group they were placed in. The case description (Appendix B) was based on a hotel reservation scenario and was adopted from Evermann and Wand (2006). Subjects verbalized while developing the UML class diagrams and their verbalizations were recorded for analysis. Sample models developed by subjects (one from each group) are show in Figure B1. The focus of the ontological guidelines was on using mutual properties to model interactions. The groups untrained on ontological guidelines were not exposed to the concept of mutual properties.

Results

First the examples of both types of breakdowns faced by the subjects are presented in Table 4.

Table 4: Examples of breakdowns of subjects

Subject ID	Example of Explicit breakdowns	Example of Implicit breakdowns
Q2B	*"So memberships belong to hotel or guest?" "Invoice should be connected to guest, no, I don't think so"*	*Adds "makesreservation ( ) and deletesreservation ( ) to the Hotel class and then deletes these operations.*
Q3 D	*"actually I need to come up with another association class which am not understanding..."*	*"the operations of employee can be like reserve room..hmm...provide info...guest class is"*
Q3B	*oh sorry sorry...scheme number should be attribute of membership but not for hotel guest*	*"ok...membership is association class between hotel guest and privileged hotel guest...room reservation..."*

Q2 and Q3 -refers to the quadrants 2 and 4 in Table 3 respectively and the alphabets B and D refers to subjects

Implicit and explicit breakdowns were identified in all four groups. Total number of breakdowns during the entire modeling task and breakdowns related to attempts made when developing association classes are provided in Table 5. As most of the guidelines were focused on modeling association classes, breakdowns related to attempts to model association classes were also identified.

Table 5: Breakdowns during developing classes and association classes

Subject	Breakdowns-total			Breakdowns-association class		
	Explicit	Implicit	Total	Explicit	Implicit	Total
Q1A	0	5	5	0	3	3
Q1B	1	7	8	0	1	1
Q1C	1	3	4	0	0	0
Q1D	1	4	5	1	1	2
Q1E	0	3	3	0	0	0
Average	**0.6**	**4.4**	**5.0**	**0.2**	**1.0**	**1.2**
Q2A	2	6	8	1	2	3
Q2B	2	6	8	1	1	2
Q2C	4	8	12	1	3	4
Q2D	1	8	9	0	3	3
Q2E	2	4	6	1	1	2
Average	**2.2**	**6.4**	**8.6**	**0.8**	**2**	**2.8**
Q3A	3	6	9	2	3	5
Q3B	3	6	9	3	2	5
Q3C	8	6	14	6	3	9
Q3D	4	6	10	3	4	7
Q3E	4	8	12	3	4	7
Average	**4.4**	**6.4**	**10.8**	**3.4**	**3.2**	**6.6**
Q4A	0	5	5	0	0	0
Q4B	1	4	5	0	1	1
Q4C	0	3	3	0	0	0
Q4D	1	5	6	0	0	0
Q4E	0	4	4	0	1	1
Average	**0.4**	**4.2**	**4.6**	**0**	**0.4**	**0.4**

Q-refers to the quadrants in Table 3 and the alphabets (A-E) refers to subjects

First, we note that the average number of breakdowns for groups Q1 and Q4 are similar (5.0 and 4.6) but the number of breakdowns for modeling association class is higher in Q1 than in Q4. However, only few association classes were developed by subjects in Q1, whereas all subjects in Q4 developed association classes. Second, the number of breakdowns for groups Q2 and Q3 are high. In particular, subjects in Q3 had more breakdowns than subjects in Q2. This indicates that it is more problematic to model association class when modeling guidelines are provided but no mutual properties are given. As subjects in Q3 had more breakdowns in modeling association classes, this indicates that these subjects attempted to create association classes because they were provided with guidelines. The average number of breakdowns in Q2 and Q3 combined is higher than in Q1 and Q4 combined (9.7 as compared to 4.8). This data supports our proposition that modelers placed in quadrants 1 and 4 will have fewer breakdowns when developing the UML class diagrams in comparison to the modelers placed in quadrants 2 and 3.

Several interesting observations emerged from this study. First, the average number of cognitive difficulties of subjects in Q4 was similar to those in Q1. As the quality of the models developed by subjects in Q4 is expected to be higher than those developed by subjects in Q1, we suggest that providing modelers with guidelines and complete description is better than providing no guidelines at all. However, modelers in Q4 might still face cognitive difficulties if they are not adequately trained to understand and apply the modeling guidelines. Second, when comparing the breakdowns in all four groups we found that the modelers in Q4 had the lowest number of breakdowns. This suggests that the proper use of guidelines can significantly reduce cognitive efforts of modelers. Finally, we conclude that providing guidelines without complete domain information (as in Q3) is worse than providing no guidelines at all (as in Q1) or providing complete domain descriptions only (as in Q2).

CONCLUSION

Modeling guidelines are designed to help modelers develop conceptual models. Thus it is assumed that these guidelines assist modelers in developing conceptual models. However research suggests that modeling guidelines are complex. This paper compared several modeling situations and tested the effects of these situations on modelers' cognitive processes. In summary, our study provided evidence that modeling guidelines should be developed and used by modelers whenever possible. However, to face less cognitive difficulties while modeling, modelers must select and use guidelines carefully.

Although we provided some evidence to support our proposition, more empirical testing needs to be done in this area. For example, similar studies can be replicated with different guidelines developed from other theoretical approaches. We plan to continue testing our proposition using different modeling languages (such as ER Diagrams). We also intend to extend this study by analyzing the quality of the models developed by the subjects in each group. Modeler's modeling efforts can then be correlated with the quality of the models developed.

APPENDIX A

Table A1: Guidelines that should not be violated while modeling association classes (adapted from Evermann and Wand, 2005)

1)	An association class must not possess methods or operations
2)	An association class must possess at least one attribute
3)	An association class must not be associated with another class
4)	An association class must not participate in generalization relationships
5)	An association class represents a set of mutual properties arising out of the same interaction between class instances. Different sets of mutual properties must be modeled in different associations

APPENDIX B: EXPERIMENTAL MATERIALS AND SAMPLE MODELS DEVELOPED

Table B1: Brief descriptions of UML Class diagram concepts [provided to all four groups] (adapted from Hoffer et al., 2007)

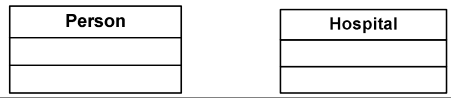
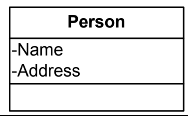
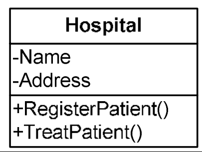
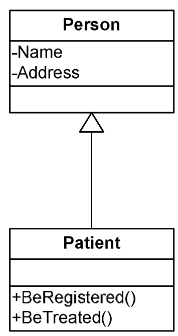
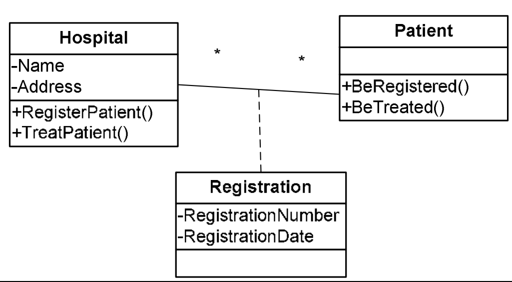
Concept	Definition	Example
Class	a class is a set of objects that share the same properties and/or behaviors	**Person** and **Hospital** are concepts and therefore are modeled as classes 
Attributes	are properties held by members of a class. Attributes can have constant (such as DateOfBirth) or variable values (such as Address)	The Person class can have Name and Address as attributes. 
Operations	are functions or services that are provided by all the instances of a class to invoke behavior in an object	The two operations of the Hospital class are register patients and treat patients. 
Subclasses	a sub-class has more attributes or more operations than the general class	A Patient is a subclass of a Person 
Association	is the relationship among instances of classes	Hospital and Patient are related as Hospital Treats patients 
Association class	an association that has attributes or operations of its own	Registration is an association class that has attributes RegistrationNumber and RegistrationDate 

Table B2: Training on modeling UML classes [provided to the groups who did not use the ontological guidelines]

Develop a UML class diagram based on the following information:

"*By registration, a person becomes a patient of a hospital.*"

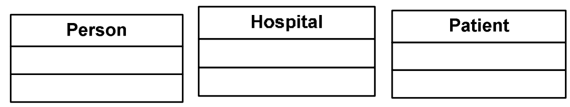
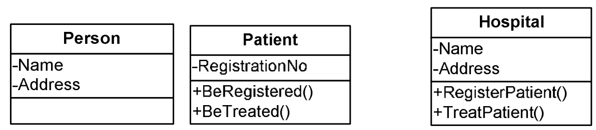
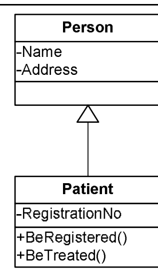
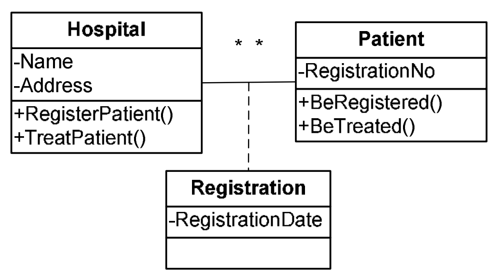
Steps	Examples
Creating classes • Identify the main concepts from the description: Person, Patient, and Hospital • Create classes named **Person**, **Patient**, and **Hospital**	
Creating attributes and methods • The class **Person** can have attributes - **Name** and **Address** • The class **Patient** can have attribute - **RegistrationNo** and methods such as **BeRegistered** and **BeTreated**. • The class **Hospital** can have attributes such as - **Name** and **Address** and methods such as **RegisterPatient** and **TreatPatient**	
Creating subclasses • Make **Patient** a subclass of **Person** • The attributes and methods of the main class are inherited by the subclass. Therefore a subclass must have more attributes or operations that the general class. **Patient** class does not have **Name** and **Address** as attributes.	
Creating Association class • Identify associations and model them as association classes. **Registration** is an association class between **Hospital** and **Patient** • Identify the attributes of association classes. **RegistrationDate** can be an attribute of the association class **Registration**.	

Table B3: Description of concepts [provided to the groups trained with guidelines]

Concept	Definition
Substantial entities	are things that one believes to exist or might exist in the world. A student is a physical thing thus considered as substantial entity. On the other hand, the skill of a student is not a substantial entity. Similarly, while customer is a physical entity, customer account is not a physical entity.
Mutual properties	substantial entities possess intrinsic or mutual properties. Intrinsic properties are possessed by the thing alone whereas mutual properties are shared by two or more different things. For example, height is an intrinsic entity but salary is a mutual property of two substantial entities- employee and a company.

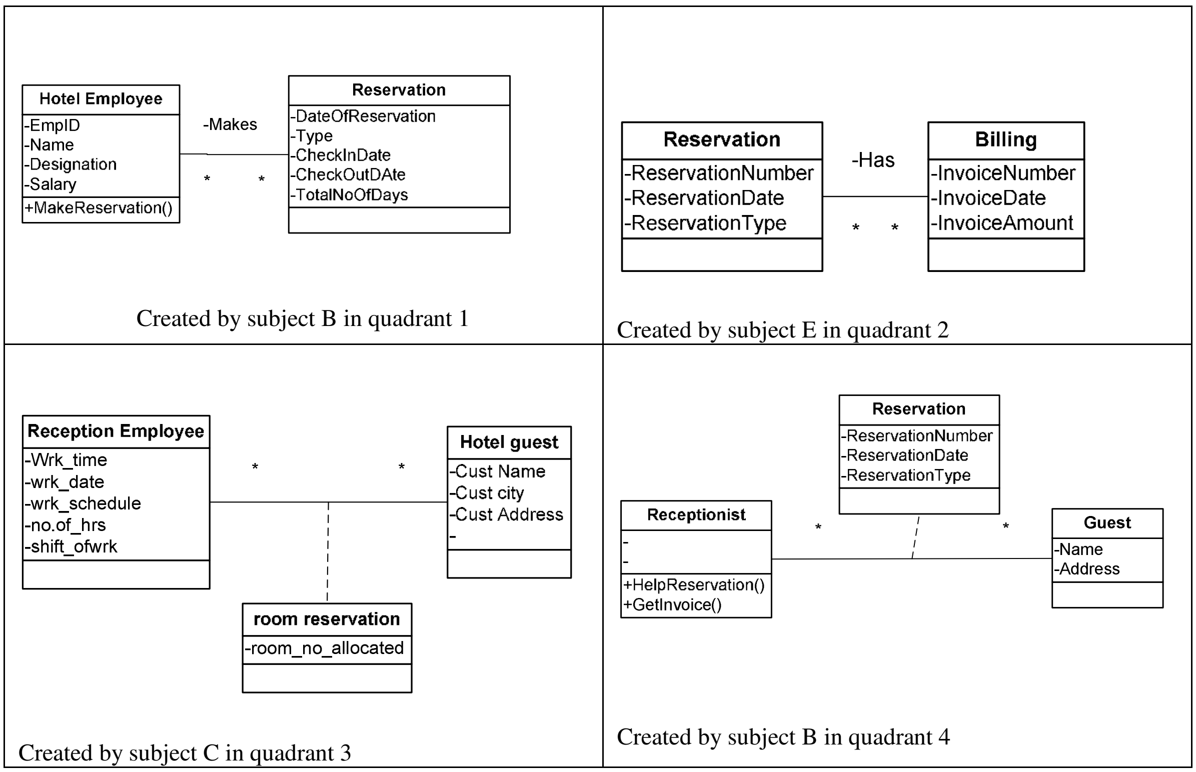
Case Description

Prepare a UML class diagram in MS Visio using the following description. The attributes mentioned below should be used to develop some of the classes.

A reputed hotel is located in a major US city. On arrival of the guests, the hotel reception employees help the guests to get room reservation. As part of the hotel promotion, hotel guests can become a privileged hotel guest which will provide the guests with some additional benefits. By accepting the hotel's membership, a guest becomes a privileged hotel guest. A privileged hotel guest gets a complementary pick up service by the hotel car. At the end of the stay, the reception employees hand over the invoice to the guests.

List of attributes

ReservationNumber, ReservationDate, ReservationType, InvoiceNumber, InvoiceDate, InvoiceAmount, MembershipNo, MembershipStartDate, MembershipExpiryDate, ScheduleDate, ScheduleTime, and ScheduleLocation.



Created by subject B in quadrant 1

Created by subject E in quadrant 2

Created by subject C in quadrant 3

Created by subject B in quadrant 4

Figure B1: Example of parts of models created by subjects

REFERENCES

1. Bera, P., Burton-Jones, A., and Wand, Y. (2011) Guidelines for Designing Visual Ontologies to Support Knowledge Identification, *MIS Quarterly* (*forthcoming*)
2. Bera, P., Krasnoperova, A., & Wand, Y. (2010) Using OWL as a Conceptual Modeling Language. *Journal of Database Management* 21, 1, 1-28.
3. Bunge, M. (1977). Ontology I: The Furniture of the World (Vol. 3). New York: D. Reidel Publishing.
4. Burton-Jones, A. & Meso, P. (2006) Conceptualizing systems for understanding: An empirical test of decomposition principles in O-O analysis. *Information Systems Research* 17, 1, 38-60.
5. Davies, I., Green, P., Rosemann, M., Indulska, M. & Gallo, S. (2006). How do practitioners use conceptual modeling in practice? *Data & Knowledge Engineering*, 58, 358-380.
6. Ericsson, K. A., and Simon, H. A. (1984). Protocol Analysis: Verbal Reports as Data. Cambridge, Mass.: MIT Press.

7. Evermann, J., & Wand, Y. (2005). Ontology Based Object-Oriented Domain Modelling: Fundamental Concepts. *Requirements Engineering Journal,* 10, 2, 146-160.

8. Evermann, J. and Wand, Y. (2006) Ontological Modelling Rules for UML: An Empirical Assessment, *Journal of Computer Information Systems*, 46, 5, 14-19

9. Fettke, P. (2009) "How Conceptual Modeling Is Used" *Communications of the Association for IS,* 25, 1.

10. Guizzardi, G., Wagner, G., Guarino, N, and Sinderen, M. (2004) An Ontologically Well-Founded Profile for UML Conceptual Models. *Proceedings of the 16th International Conference on Advanced Information Systems Engineering* (CAiSE), Riga, Latvia, June 7-11, 112-126

11. Hoffer, J.A., Prescott, M.B. McFadden, F.R. (2007) *Modern Database Management*. 8th ed. Pearson Prentice Hall, Upper Saddle River, N.J.

12. Milicev, D. (2007) On the Semantics of Associations and Association Ends in UML, *IEEE Transactions of Software Engineering,* 33, 4, 238-251.

13. Newell, A., and Simon, H. (1972). *Human Problem Solving*. Englewood Cliffs, NJ: Prentice Hall.

14. Siau, K. and Loo, P. (2006) Identifying Difficulties in learning UML, Information *Systems Management*, 23, 3, 43-51.

15. Soffer, P. and Hadar, I. (2007). Applying Ontology-Based Rules to Conceptual Modeling: A Reflection on Modeling Decision Making, *European Journal of Information Systems*, 16, 599-611

16. Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive Science* 12, 2, 257–285

17. Vessey, I., and Conger, S. (1994). Requirements specification: learning object, process, and data methodologies. *Communications of the ACM*, 37, 5, 102-113.

18. Weber, R. (2003) "Conceptual Modelling and Ontology: Possibilities and Pitfalls," *Journal of Database Management,* 14, 3, 1-20.