Usability Assessment of a UML-based Formal Modelling Method

Rozilawati Razali, Colin Snook, Michael Poppleton, and Paul Garratt

Dependable Systems and Software Engineering, School of Electronics and Computer Science, University of Southampton, SO17 1BJ United Kingdom

{rr04r, cfs, mrp, pwg}@ecs.soton.ac.uk

Abstract. Conceptual models communicate the important aspects of a problem domain to stakeholders. The models therefore should be accessible to users who need to interpret them. On the other hand, the quality of the produced models is highly dependent on the usability of the modelling method used. This paper presents a series of usability assessments conducted on a method that integrates the use of a semi-formal notation, namely the Unified Modelling Language (UML) and a formal notation, namely B. The assessments included a controlled experiment that evaluated the comprehensibility of the produced model and a survey that assessed the modelling process. The results suggest that the method is able to produce a comprehensible model. The method is accessible to users when the principles and roles of each notation are obvious and well understood, and when there is strong support from the environment.

1 Introduction

Conceptual models are vital in the development and maintenance of software systems. They allow the characteristics of the existing and future systems to be captured and understood. A conceptual model is produced through the use of a designated modelling notation. Some examples of the existing notations include semi-formal notations such as Entity-Relationship Diagram (ERD) [1] and Unified Modelling Language (UML) [2], and formal notations such as Z [3] and B [4]. In addition, there are also notations that integrate both semi-formal and formal such as UML and Z [5].

Formal notations use mathematical symbols and interpretation to describe a system. Thus, they have the ability to increase a model's precision and consistency, which is necessary especially for critical systems [6]. Formal notations however are regarded as being difficult to comprehend [7]. In contrast, semi-formal notations employ graphical symbols. It is therefore perceived as more accessible but they cannot be verified systematically to ensure a model's accuracy. By integrating formal and semi-formal notations, it may be that practitioners can produce a model that is accurate, consistent and more accessible to them. One possible approach to this integration is to combine the formal notation of B and the semi-formal notation of UML. A method called UML-B [8] is one such product. The rationale of this integration is that B has strong industrial supporting tools such as Atelier-B [9] and B-Toolkit [10], and UML has become the de facto standard for system development [11].

This paper presents an investigation into the usability of the UML-B method. Usability in this context means the understandability/comprehensibility, learnability, operability and attractiveness of the method. The objective of the investigation was to assess whether or not the method could produce a model that is usable to users who interpret the model, and support developers during modelling process. The investigation comprised a controlled experiment and a survey. The controlled experiment evaluated the comprehensibility of the produced model while the survey assessed the usability of the method as a whole. This paper summarises the controlled experiment but its main contribution is the usability assessment based on the survey. Section 2 of the paper provides an overview of the experiment while Section 3 presents the survey. Section 4 concludes and describes the future work.

2 Controlled Experiment

The experiment evaluated the notation used in the UML-B method in order to explore whether it could improve model comprehensibility. The evaluation was based on the comparison made between a UML-B model and a B model. A UML-B model comprises the semi-formal notation used in the UML, namely class and statechart diagrams, and the formal notation used in the B method, namely B syntax. A B model comprises only the B syntax. The B syntax in general contains mathematical constructs based on set theory, relations and predicates. The measurement used in the evaluation focused on the efficiency in performing the comprehension task, that is, accuracy over time. The following paragraphs briefly explain the experiment. The detailed elaboration of the experiment can be found in [12].

2.1 Objectives and Hypotheses

The experiment was conducted to confirm (or refute) a theory that suggests the notation used in the UML-B model has a particular effect on the users, making it better in some way than the notation used in the B model. The experiment attempted to answer the following research question:

Is a UML-B model easier to understand (i.e. efficiency in understanding and performing the required tasks) than a B model for practitioners with limited hours of training?

The hypotheses used were:

Null hypothesis (H_0): The UML-B model is no more comprehensible than the B model
Alternative hypothesis (H_1): The UML-B model is more comprehensible than the B model

2.2 Execution

Forty-one subjects participated in the experiment. They were third-year Undergraduate and Masters students. The experiment employed a related within-subject design where each of the subjects was trained and assigned a task on both models. Since there were two treatments to be tested in the experiment, the subjects were allocated randomly into two groups using blocking and balancing techniques. The experiment was a cross-over trial [13], which was used to eliminate task direction bias and ability effect. At one session, one group of subjects was assigned a task on the UML-B model while the other was assigned the same task on an equivalent B model. The reverse was then carried out in the subsequent session. The measured comprehension criteria include the interpretation of the symbols used, the tracing of input and output, the mapping between models and problem domains, and the modification task on the models. The response variables were score (accuracy) and time taken to answer the questions. The score and the time taken were used to determine the measure of efficiency, that is, rate of scoring (score over time taken).

2.3 Results and Analysis

There were two types of comprehension measurement and analysis; *overall comprehension task* and *comprehension for modification task*. The experiment employed a robust statistical method called bootstrap methods and permutation tests for the statistical inference [14]. The analysis considered the period effect [13], which is the chance of detecting effects due to the session when the treatment is applied. The results suggest with 95% confidence that a UML-B model could be up to 16% (*overall comprehension*) and 50% (*comprehension for modification task*) easier to understand than the corresponding B model.

The UML-B model was commented on by the subjects as being easier to understand the scenario and the relationships between operations more quickly, easy to develop especially on computers and more logical to developers. Nevertheless, the model was said to be useful only with good tool support. The UML-B model was also commented as being quite ‘messy’ since the information was scattered around the class and statechart diagrams.

3 Survey

The controlled experiment evaluated the notation comprehensibility in terms of how easy it is to understand a UML-B model from the perspective of users who interpret the model. The results of the experiment suggest that the UML-B model is more comprehensible than the B model. The findings however cannot suggest by any means that the notation is also usable from the perspective of developers who use the UML-B method for modelling. Neither could they determine whether or not the notation suits the developers’ common needs and expectations.

The following paragraphs present a survey conducted on the UML-B method, particularly the notation used. The notation used in the method includes the use of class and statechart diagrams of UML and the use of B syntax for expressing constraints and actions on the diagram elements. The survey assessed the usability of the notation from developers' perspective. As usability depends on the notation and its environment, the evaluation included the tools that accompany the method namely Rational Rose [15] and U2B [8], whenever appropriate. Rational Rose provides the environment for the UML-B model development while U2B is a tool that generates a B model from a UML-B model so that it can be verified by the B tools such as Atelier-B and B-Toolkit.

3.1 Objectives and Methodology

The survey was qualitative in nature. Despite the fact that some of the data were quantified using an ordinal scale, the bulk of the analysis was interpretative. This type of analysis was carried out due to the problem at hand, that is, the survey attempted to understand the nature of experience of using the UML-B method. The survey adopted one approach to dealing with qualitative data employed in the social sciences; the grounded theory [16-17]. The theory in the approach means theory that is derived from data, systematically gathered and analysed through the process. The approach was chosen because it allows the study to be initiated without a preconceived theory in mind, where the researchers could start with a phenomenon and allow the theory to emerge from the collected data. As the theory is drawn from data, it is likely to offer insight, enhance understanding and provide a meaningful guide to action.

The survey aimed to formulate tentative theories of the usability of integrated methods (semi-formal and formal notations) such as UML-B, based on the understanding obtained from the qualitative analysis using the grounded theory approach. As one single study can never embrace all possible situations, the survey sought to provide some preliminary evidence of the method's likely strengths and weaknesses when used under certain known conditions. It was also intended to identify any threats that could hinder the method's usability and any opportunities that could improve the method further. The tentative theories could act as a basis for further investigation and analysis in future.

One of the subjective comments obtained from the controlled experiment was that the UML-B was seen as easy to develop particularly on computers (see Section 2.3). The method was also commented to be useful only with good tool support. The hypotheses were given by subjects who dealt with the already developed UML-B model, not modelling. This could not suggest that the hypotheses are true from developers' perspective for modelling purposes. The survey therefore included these hypotheses in its investigation of the phenomenon through the following broad questions:

Do individuals who develop a model using the UML-B method perceive them (i.e. method and model) as usable (easy to understand, easy to learn, easy to operate, and attractive)?

What are the characteristics of UML-B method and UML-B model that affect their usability?

3.2 Materials

The survey instrument was developed based on the ideas proposed in the Cognitive Dimensions of Notations (CD) usability framework [18]. The framework was adopted because it is a tool that aids the usability evaluation of information-based artefacts. As a usability tool, it captures a significant amount of psychology and Human Computer Interaction (HCI) aspects that focus particularly on the notational design. The framework comprises fourteen dimensions as illustrated in Table 1 below, which acted as the response variables in the survey.

Table 1. The Cognitive Dimensions

Dimension	Description
Abstraction Gradient	Level of grouping mechanism enforced by the notation
Closeness of Mapping	Mapping between the notation and the problem domain
Consistency	Similar semantics are presented in a similar syntactic manner
Diffuseness	Complexity or verbosity of the notation to express a meaning
Error-proneness	Tendency of the notation to induce mistakes
Hard Mental Operations	Degree of mental processes required for users to understand the notation and to keep track of what is happening
Hidden Dependencies	Relationship between two entities such that one of them is dependent on the other but the dependency is not fully visible
Premature Commitment	Enforcement of decisions prior to information needed and task ordering constraints
Progressive Evaluation	Ability to evaluate own work in progress at any time
Provisionality	Flexibility of the notation for users to play with ideas
Role-expressiveness	Purpose of an entity and how it relates to the whole component is obvious and can be directly implied
Secondary Notation	Ability to use notations other than the official semantics to express extra information or meaning
Viscosity	Degree of effort required to perform a change
Visibility/ Juxtaposibility	Ability to view every component simultaneously or view two related components side by side at a time

The questions for the survey were constructed by following the proposed CD questionnaire [19]. The advantage of using the standard instrumentation such as proposed by the CD questionnaire is that it has been assessed for validity and reliability by the authors. As the CD framework is widely used by other researchers who are investigating the usability of notations such as [20-21], it also provides a mechanism to compare the results of this survey with the results of other similar studies.

The CD questionnaire is intended to present the dimensions in general terms, applicable to all information artefacts rather than presenting descriptions specialised to a specific system under consideration. The questionnaire was therefore tailored and modified slightly to reflect the characteristics of the UML-B method. The proposed CD questionnaire also employs an open-ended question approach, which could complicate the data analysis. The questions for the survey were thus designed to include a set of answers using an ordinal scale together with the open-ended questions. Besides reducing the data analysis's complexity, it allows the survey to obtain some quantitative measures rather than qualitative measures entirely.

In addition to the CD framework, the questions of the survey were also constructed based on the usability criteria proposed by the International Organization for Standardization (ISO) [22-23]; understandability, learnability, operability and attractiveness. There were twenty questions in the survey; fourteen questions reflected the fourteen dimensions of the CD framework, five questions represented the ISO's usability criteria and one question gathered suggestions for improvement. The questions in the survey were presented in random order without following a specific sequence of dimensions. To ensure the questions were purposeful and concrete, the general guidelines on survey question construction were followed [24].

The questions used an ordinal scale that provided the respondents with five possible levels of agreement such as *–2* for *Very Difficult* to *2* for *Very Easy*. There were also questions that required either *Yes*, *No* or *Not Sure*. The five levels were chosen because they cover the possible categories of the variables. An odd number of levels were used because odd numbers contribute to the achievement of better results as they are balanced. Besides the selection on the scale, justification of the answer given was also required such as *Why?* or *Which part?*. This acted as the qualitative data, which were used together with the quantitative data on the scale for the analysis. The survey questions and raw data can be found in [25]. To give an overview of the questions, below are some examples of the survey questions. The first question concerns the "Visibility and Juxtaposability" dimension, which also relates to the "Operability" criteria of ISO. The second question involves the "Hard Mental Operations" dimension that also implies the ISO's "Understandability" criteria.

If you need to compare different parts of your UML-B model (e.g. between diagrams or windows of different operations etc.), how easy is it to view them at the same time in Rational Rose?

Very Difficult				*Very Easy*
-2	*-1*	*0*	*1*	*2*

Why?

Do you find any complex or difficult tasks to work out in your head when modelling your UML-B model?

No *Not Sure* *Yes*

If Yes, what are they? If No or Not Sure, why?

The CD framework describes the necessary conditions for usability based on the structural properties of a notation, the properties and resources of an environment, and the type of user activity; *incrementation*, *transcription*, *modification*, *exploratory design*, *searching* and *exploratory understanding* [26]. In particular, it addresses the question whether the users' intended activities are adequately supported by the structure of the notation used and its environment. For the survey, the identified users' intended activity was *exploratory design*, which the users employed the UML-B method (notation and environment) to design a conceptual model. The survey questions and analysis therefore were tailored to focus on this aspect.

The survey questions were reviewed by a focus group prior to distribution. There were four people involved in the process. The purpose of the review was to identify any missing and unnecessary questions as well as ambiguous questions and instructions.

3.3 Participation

Ten participants responded to the survey. They were Masters students of Software Engineering course at the University of Southampton, who registered for the "Critical System" course in Spring 2006. They were selected as the samples because they were taught formally on the B method (nine hours) and the UML-B method (one hour) during the course. Basic knowledge of both methods is necessary to develop a UML-B model. The participants had some practical experience of using the UML-B method and its tools when participating in the survey, that is, they used them to develop a model of a system in one of the coursework at the end of the course.

The survey adhered to the University's ethical policies and guidance for conducting research involving human participants. The participants were aware that the survey was intended for research purposes. They were motivated to participate as it helped them in exploring the method besides providing a space for reflection on the learning prior to the examination.

The subjects were in the final semester of their Masters course. They therefore had reasonable amount of experience and knowledge of software development. Some of them had some work experience. They were the next generation of professionals, thus they represented closely the population under study; software developers.

3.4 Results and Analysis

The survey adopted the grounded theory approach for the data analysis. In addition to obtaining understanding of the use of the UML-B method, the survey aimed to formulate tentative theories of the usability of such integrated methods in general. The theory in the approach denotes a set of discrete categories that are systematically connected through statements of relationship. The categories in essence are abstract concepts that describe the phenomenon under study whereas the statements of relationship are the interrelated properties of those categories.

The analysis focused on how often the categories emerged in the data under varying conditions. The idea was to form a theoretical framework, thus the analysis involved the generation of general categories or concepts rather than specific to any individual cases. For example, issues of using Rational Rose and running U2B were conceptualised as *Availability and usefulness of supporting tools*. The analysis did not intend to specifically delineate every single limitation of the tools. Rather, the objective was to identify and propose a set of categories that can be used as a basis for examining the usability of other similar methods or testing the theories in future.

The properties for the categories were derived by having queries such as *what*, *why*, *how* and *when* during the analysis process. For example, respondents mentioned the issue of learning the UML and B several times in their answers. Therefore, *Learnability of notations and tools* was recognised as one of the categories. On the other hand, it is necessary to know *what* aspect of the notations and their tools that was easy and difficult to learn, *when* and *why* they happened, in order to understand the phenomenon. To answer the queries, evidence was obtained and accumulated from various parts of the questionnaire. This included both the quantitative (ordinal scale) and qualitative (subjective) data. The use of CD framework that shapes the dimensions of usability investigation facilitated the identification of the categories and properties.

The following paragraphs list the categories and elaborate their properties. The properties were mainly formulated based on the stated subjective comments, which were also supported by the data on the ordinal scale. The corresponding data (reasoning based on CD) that support the statements are stated in the parentheses in the paragraphs, which details can be found in [25].

Model Structure and Organisation. The UML portion of the UML-B method allows the system properties and behaviours to be illustrated using the class and statechart diagrams. Each diagram represents the system from a specific perspective. For example, the class diagram shows the attributes and relationships between entities in the system while the statechart diagram delineates the states and transitions involved in the system operations. In modelling a UML-B model, the users employ the diagrams to illustrate the system properties from these perspectives.

The diagrams are equipped with formal semantics where the characteristics and behaviours of the systems are specified more precisely. Formal semantics in the form of B syntax are added at different parts of the diagrams so that they can be transformed to a B model. For example, the global definition such as variables declaration and invariants are placed at the class diagram level while the conditions and effects of the behaviours are placed at the statechart diagram level. Despite being scattered at several parts of the model, the method has the ability to transform the diagrams and consolidate the semantics as a single B model through its tool, namely U2B.

Despite being logical, having the formal semantics at different parts of the model causes an accessibility issue to the users. They need to switch around different parts of the model to specify the formal semantics. Rational Rose supports the display of multiple windows at one time. However, having to deal with several displayed windows simultaneously in Rational Rose seems to be a problem that often causes confusion (Reasoning: “Visibility and Juxtaposibility” dimension). The users have to view not only the windows that display the class and statechart diagrams but also the pop-up windows that carry the semantics for each of the diagrams. In fact, some of these windows have to be on top of each other due to the limited screen space. This leads the users to overlook certain aspects of the model and prone to errors (Reasoning: “Error Proneness” dimension). The users can view and subsequently check the model using the B tools by translating it to a B model using U2B at any modelling stage they like (Reasoning: “Progressive Evaluation” dimension). However, having to transform the model particularly during formulating and synthesising ideas has been regarded as a ‘noise’. In addition, the model transformation at early stages where many aspects of the model have yet to be given careful thought will generate error messages in the B tools. Starting modelling with many generated errors can be a daunting experience especially to new users.

This finding supports the comment obtained from the controlled experiment where the UML-B model had been regarded as ‘messy’ (see Section 2.3). The ‘messiness’ is not only caused by the scattered information but also the display of multiple windows at a time. The structure of the model does affect its accessibility for both model reading and development, even on the computer screen. The cognitive psychology theory underpins this phenomenon is that humans have limited amount of information that can be processed at one time, which the way material is organised and presented has an effect [27]. When the related information is separated from each other on the page or screen, users have to use cognitive resources to search and integrate it. Users are less likely to be able to hold the separated information in working memory at the same time especially if the information has a high intrinsic cognitive load [28]. In general, formal notation such as B syntax is high in intrinsic cognitive load because it involves concurrent interactions between its syntactical and semantic characteristics.

As a UML-B model always involves the use of more than one UML diagram that carries the respective B syntax, the issue of scattered information is seen as unavoidable. However, the effect of split-attention can be reduced if the modelling tool allows the switching and viewing different parts of the model more conveniently and less distracting.

Availability and Usefulness of Supporting tools. Rational Rose and U2B are the main supporting tools in the UML-B method. These tools have been useful in some aspects (Reasoning: "Consistency" dimension; "Secondary Notation" dimension; "Utility of U2B"). On the other hand, there are also several user-friendliness issues discovered by the users. For example, Rational Rose does not support some changes automatically, which causes the modification process to be unnecessarily tedious (Reasoning: "Viscosity" dimension). If a variable name is changed in the class diagram, the change is not reflected in other parts such as in the statechart diagram or in the semantics where the variable name is used. A similar situation applies to variable deletion. Thus, the changes have to be done manually by visiting the respective parts of the model.

U2B in general has received a fairly good acceptance among the users. This is due to its obvious role, that is, to transform the UML-B model to a B model. By executing several simple steps, the users can generate a B model and execute the verification task using the B tools. This is the reason why the tool is seen as easy to learn and use (Reasoning: "Learnability of U2B"). The automatic transformation has alleviated some pains that would occur when modelling a B model from scratch. At the very least, it provides basic structures for the B model where the users could extend further by adding more details. The simplicity of U2B however has made the verification task remains in the B tools. No matter how simple, U2B or even Rational Rose does not support any checking in any way. This means the users have to transform the UML-B model to a B model and run the B tools each time they change ideas even it involves only a minor change. Otherwise, there is no way they could be sure whether or not the changes are acceptable. The generated B model will contain numerous types of errors from the simplest to the complex ones, which can only be realised during the model verification using the B tools. Because of this reason, the users feel that the method is not good enough for playing around with ideas (Reasoning: "Provisionality" dimension). Some simple checking such as unused variables and typing errors of B syntax at the modelling and transformation levels would be useful to the users. This can act as the front line checking, which eliminates trivial errors before pursuing more extensive verification in the B tools. Rather than introducing all types of errors at once, evolutionary phases of checking could make the verification task less daunting and troublesome to the users. As the tool lacks of these elements, it does not fully meet the users' expectation (Reasoning: "Utility of U2B").

This finding supports the comment obtained from the controlled experiment where several subjects in the experiment believed that the method is useful only with good tool support. Although the necessary tools are available, there are several aspects that should be improved in order to increase their utility (Reasoning: "Future Improvement"). Perhaps a more seamless modelling environment should be created so that users do not have to perform several individual and intricate steps during modelling.

Learnability of Notations and Tools. The successful use of the UML-B method relies on the fact that users have to be familiar with UML and B. Otherwise, the integration of both notations could not be understood or valued. It has been found that it is difficult if not impossible to obtain the understanding of the notations used in both UML and B at the same time (Reasoning: "Learnability of UML-B method"). Even though the users have been exposed to UML and B for some time, some mental burden still occurs during the process (Reasoning: "Hard Mental Operations" dimension). Having to think, integrate and harmonise two styles of modelling from two different methods seems to be problematic.

The model transformation provided by U2B also requires some learning (Reasoning: "Learnability of UML-B method"). A UML-B model in essence carries two types of semantics; explicit B syntax specified by the users in the UML diagrams that U2B transforms as it is in the B model, and implicit B syntax that U2B implies and generates automatically from the diagrams. For example, behaviours of the operations have to be specified by the users using the B syntax in the UML diagrams whereas classes and associations in the diagrams are translated automatically as the respective sets and variables in the B model. The users have to understand these transformations and why they are accomplished in such ways (Reasoning: "Learnability of U2B"; "Hidden Dependencies" dimension) as it affects the way they should do the modelling (Reasoning: "Closeness of Mapping"). Moreover, learning on how to do modelling in Rational Rose is also required (Reasoning: "Learnability of UML-B method").

Modelling the UML diagrams is regarded as quite straightforward (Reasoning: "Role Expressiveness-Diagram" dimension; "Error Proneness-Diagram" dimension), which ease the process of describing what is intended (Reasoning: "Diffuseness" dimension; "Closeness of Mapping" dimension). Despite the fact that B modelling imposes some task ordering and requires users to define and group things beforehand, the diagrams have somehow diluted the effects (Reasoning: "Premature Commitment" dimension; "Abstraction Management" dimension). Perhaps these factors help to explain why a UML-B model is seen as more approachable than a B model and thus, the UML-B method is preferred for formal modelling (Reasoning: "Method attractiveness").

On the other hand, specifying the UML diagrams with the correct formal semantics is perceived as hard and error-prone (Reasoning: "Error Proneness-Syntax" dimension; "Hard Mental Operations" dimension). Shallow understanding of how the formal semantics should work with the UML diagrams, lack of comprehensive documentation on the method (Reasoning: "Usefulness of Documentation") and the need to grasp the underlying principles of the participated methods and tools mentioned above have downgraded the operability of the method (Reasoning: "UML-B method's operability"). To attract new users to the method, a more comprehensive documentation should be readily available (Reasoning: "Future Improvement"). The documentation should cover more on the practical aspect of the method and its tools (Reasoning: "Usefulness of documentation"), rather than just theory. Currently, the available documentation on the method is not helping the users much in this aspect (Reasoning: "UML-B method's accessibility")

Functionality of Notations. Rational Rose provides specification windows in each diagram for specifying the semantics. There are two types of diagrams involved in the UML-B method, thus the users are provided with two types of specification windows. One is in the class diagram and the other is in the statechart diagram. Regardless the location, U2B is able to extract the semantics and treat them accordingly as a B model.

The semantics in the statechart diagram are transformed as a nested condition under the primary condition, which is obtained from the class diagram. In many cases, the semantics of the statechart diagram can also be placed directly in the specification windows of the class diagram. As far as the users know what the states and transitions involved in the operations, they can specify it literally as a series of conditions in the specification windows of the class diagram. Despite providing an alternative in modelling, the flexibility somehow has made the role of the semantics in the statechart diagram or even the statechart diagram unclear to some users (Reasoning: "Role Expressiveness-Diagram" dimension; "Role Expressiveness-Syntax" dimension). The users seem to prefer specifying the whole semantics in the class diagram, as it is more obvious and straightforward. It could also reduce the mental burden of having to work with two different diagrams at the same time (Reasoning: "Visibility and Juxtaposibility" dimension; "Hard Mental Operations" dimension). Moreover, the generated nested conditions from the statechart diagram tend to complicate the B model. As the end product that actually matters is the transformed B model, the users prefer to have a simple and quick solution to achieve it.

More clear roles and boundaries should be set between the formal semantics of the class diagram and the statechart diagram. The explanation on the roles and responsibilities of each part of the diagrams and semantics should be stated succinctly in the documentation, which the method is currently lacking (Reasoning: "Usefulness of documentation"). It may be better if some principles and controls can be placed on how a UML-B model should be modelled. Although it may reduce the flexibility in modelling, it can at least guide the users in modelling based on what should and should not be done rather than any way could do, which is too subjective. It can also avoid redundancy. This is particularly true for new users who mainly have no idea on how to start and pursue the modelling. Besides, the transformation of formal semantics from the statechart diagram to a B model can be smoothed further so that no unnecessary complication is introduced to users.

3.5 Discussion

The data from the survey suggest that the UML-B method is appealing to users who opt into B modelling while yet prefer working with standard development style of UML. This is particularly true when users are familiar with UML and have the capacity to appreciate what formal notations like B could offer. The graphical modelling environment alleviates the pain of developing a formal model from scratch by stimulating the idea formulation through the use of visual objects at the abstraction level. On the other hand, users are faced with the challenge of having to grasp the underlying principles of each individual notation as well as to understand how both notations work together to achieve the integration objectives. Understanding of each notation's roles and functionality in different parts of a model is required, which can easily be achieved only if the distinction between them is clear-cut. Users are also required to learn and become familiar with the individual tools that accompany each notation, which in general should provide the necessary support. In short, the survey generates the following tentative theories of the usability of integrated methods that combine semi-formal and formal notations:

Theory 1: The integration of semi-formal and formal notations requires the understanding of principles and roles of both notations as well as the rules of the integration. The principles, roles and rules ought to be obvious to users.

Theory 2: The integration of semi-formal and formal notations requires strong support from the environment. Supporting tools and comprehensive documentation should be not only available but also useful, easy-to-learn and easy-to-use.

In terms of the CD framework, goals for designing integrated methods such as UML-B were identified. The design goals were proposed based on the nature of semi-formal and formal notations, and the motivation behind the integration. Besides, they were also based on the common types of user activity involved in using such methods. In general, there are two major user activities: *exploratory design* where users use such methods to create a new model, and *modification* where users use the methods to make changes and enhancements to an existing model. Table 2 below illustrates the recommended CD profile for designing such methods. The *High* and *Low* indicate whether the dimension should be increased or reduced respectively, when such methods are designed. For example, method designers are recommended to aim at increasing *Progressive Evaluation* and reducing *Hidden Dependencies*. The *Moderate* indicates that although the dimension is desired at certain level (*High* or *Low*), it may be traded-off to suit other dimensions or the two user activities. For example, *Secondary Notation* is very useful for *modification* activity, however it may cause *exploratory design* activity difficult. Besides, the two user activities require a model to be less resistant to change (low *Viscosity*). By having *Secondary Notation*, any changes to the model can be a bit painful. Thus, it may be traded-off for achieving low *Viscosity*. It is up to method designers to decide the best compromise.

There are dimensions that specifically affect a particular notation more than the other. By integrating the notation with the other notation, it is believed that its usability can be improved (indicated by * in the Table 2). For example, it is generally known that formal notations such as B syntax involve high *Hard Mental Operations*, which causes comprehension difficult. The use of graphical representation of semi-formal notations, which is more intuitive, with the formal notations should be able to reduce the effects. Similarly, semi-formal notations in general have limited mechanisms for systematic *Progressive Evaluation*, which formal notations normally can provide. Without such interplay between the two notations, the integration effort is not worthwhile. After all, the motivation of such integrated methods is to allow one notation's limitations to be compensated by the strengths of the other.

Table 2. The proposed Cognitive Dimensions profile for designing integrated methods (semi-formal and formal notations)

Dimension	Desired level
Abstraction Gradient	Low*
Closeness of Mapping	High
Consistency	High
Diffuseness	Moderate
Error-proneness	Low*
Hard Mental Operations	Low*
Hidden Dependencies	Low
Premature Commitment	Low*
Progressive Evaluation	High*
Provisionality	High
Role-expressiveness	High
Secondary Notation	Moderate
Viscosity	Low
Visibility/Juxtaposibility	High

Note: High – to increase; Low – to reduce; Moderate – possible trade-off; * – One notation supports the other to achieve the stated desired level (otherwise, the level will be opposite)

The tentative theories and the proposed CD profile above may not be conclusive, where they can be validated and refined further in future investigations. However, they can act as the first step in understanding the nature of integrated methods such as UML-B and providing a meaningful guide to better design.

4 Conclusions and Future Work

This paper has presented empirical assessments on the usability of a method that integrates the use of semi-formal and formal notations, namely UML-B. The controlled experiment conducted on the UML-B model has provided some preliminary findings on its accessibility. The model has the ability to expedite the comprehension task where it allows the required information to be grasped more quickly. This finding is interesting as it shows that introducing some graphical features of semi-formal notation into the formal notation significantly improves the formal notation's accessibility.

The survey on the UML-B model development has indicated that the dual characteristics of the method bring several implications to users in both positive and negative ways. Combining semi-formal and formal notations allows the potential of individual notation to be strengthened while each notation's limitations to be compensated by the other. However, the integration in essence brings the loads of two individual notations, which are actually quite different in some ways. Users therefore need strong support from the environment to lessen the burden that lies beneath the integration effort. The support involves not only the tools that aid the modelling process but also resources for learning the method.

Some of the findings of the investigation are now being fed into the next generation of the UML-B method development[1]. The controlled experiment on the other hand will be replicated on different samples of the population. The measurement will be extended to include problem domain understanding, where it assesses not only the notation's ability to represent information that can be understood but also its ability to facilitate the construction of domain knowledge. Similarly, the findings of the survey can be improved further by extending the survey to a large number of users. This will help in enhancing the current understanding of the method and discovering any other factors that affect its use. The tentative theories and the proposed CD profile of integrated methods (semi-formal and formal notations) discussed in this paper can also be validated and refined further by applying them to examine other similar methods. This allows the derivation of more concrete theories and guidelines that can be used to design and improve the usability of such methods in future.

5 Acknowledgement

R. Razali thanks the students participated in the study (COMP3011-Spring 2006).

References

1. P. Chen. The Entity-Relationship Model: Toward a Unified View of Data. ACM Transactions on Database Systems, Vol.1. No.1. (1976) 9-37
2. Object Management Group. Introduction to OMG's Unified Modeling Language (UML). [Online]. Available: http://www.omg.org/gettingstarted/what_is_uml.htm (2006)
3. J. M. Spivey. The Z Notation: A Reference Manual. 2nd Edition. Prentice-Hall, Englewood-cliffs, NJ (1992)
4. J. R. Abrial. The B-Method - Assigning Programs to Meanings. Cambridge University Press (1996)
5. S. Martin. The Best of Both Worlds Integrating UML with Z for Software Specifications. Journal of Computing and Control Engineering, Vol.14. No.2 (2003) 8-11
6. M. G. Hinchey. Confessions of a Formal Methodist. In: P. Lindsay (eds): Conferences in Research and Practice in Information Technology (2002)
7. D. Carew, C. Exton and J. Buckley. An Empirical Investigation of the Comprehensibility of Requirements Specifications. International Symposium on Empirical Software Engineering (2005) 256-265.
8. C. Snook, M. Butler. UML-B: Formal Modelling and Design Aided by UML. ACM Transactions on Software Engineering and Methodology, Vol.15. No.1 (2006) 92-122
9. Steria, Aix-en-Provence, France. Atelier B, User and Reference Manuals. [Online]. Available: http://www.atelierb.societe.com/index uk.html (2007)
10. B-Core (UK) Limited, Oxon, UK. B-Toolkit, On-line manual. [Online]. Available: http://www.b-core.com/ONLINEDOC/Contents.html (2007)
11. T. Pender. UML Bible. Wiley (2003)

[1] EU Framework VI project: *Rigorous Open Development Environment for Complex Systems* (RODIN) http://rodin.cs.ncl.ac.uk/

12. R. Razali, C. F. Snook, M. R. Poppleton, P. W. Garratt, R. J. Walters. Experimental Comparison of the Comprehensibility of a UML-based Formal Specification versus a Textual One. Proceedings of the 11th International Conference on Evaluation and Assessment in Software Engineering (EASE) (2007) 1-11
13. S. Senn. Cross-over Trials in Clinical Research. John Wiley & Sons (2002)
14. B. Efron, R. Tibshirani. An Introduction to the Bootstrap. Chapman and Hall (1993)
15. IBM Software. Rational Rose. [Online]. Available: http://www-306.ibm.com/software/awdtools/developer/rose/index.html (2007)
16. B. G. Glaser, A. L. Strauss. The Discovery of Grounded Theory: Strategies for Qualitative Research. Weidenfeld and Nicolson, London (1967) (Reprint 1999)
17. A. L. Strauss, J. Corbin. Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory. 2nd Edition. Thousand Oaks, California (1998)
18. T. R. G. Green. Cognitive Dimensions of Notations. In: A. Sutcliffe, L. Macaulay (eds.): People and Computers V. Cambridge University Press, Cambridge (1989) 443-460
19. A. F. Blackwell, T. R. G. Green. A Cognitive Dimensions Questionnaire optimised for users. In: A. F. Blackwell, E. Bilotta (eds.): Proceedings of the 12th Annual Meeting of the Psychology of Programming Interest Group (PPIG) (2000) 137-152
20. M. Kutar, C. Britton, T. Barker. A Comparison of Empirical Study and Cognitive Dimensions Analysis in the Evaluation of UML Diagrams. In: J. Kuljis, L. Baldwin, R. Scoble (eds.): Proceedings of the 14th Annual Meeting of the Psychology of Programming Interest Group (PPIG) (2002) 1-14
21. E. Triffitt, B. Khazaei. A Study of Usability of Z Formalism Based on Cognitive Dimensions. In: J. Kuljis, L. Baldwin, R. Scoble (eds.): Proceedings of the 14th Annual Meeting of the Psychology of Programming Interest Group (PPIG) (2002) 15-28
22. ISO 9126-3, Software Engineering. Product Quality - Part 3: Internal Metrics. International Organisation for Standardisation (2003)
23. ISO 9126-4, Software Engineering. Product Quality - Part 4: Quality in use Metrics. International Organisation for Standardisation (2004)
24. B. Kitchenham, S. L. Pfleeger. Principles of Survey Research: Part 1-6. ACM SIGSOFT Software Engineering Notes. Vol.27. No.1-6 (2002)
25. R. Razali. UML-B Survey Questionnaires and Responses. Technical Report, Electronics and Computer Science, University of Southampton. [Online]. Available: http://eprints.ecs.soton.ac.uk/13322 (2007)
26. A. Blackwell, T. Green. Notational Systems – The Cognitive Dimensions of Notations Framework. In: J. M. Carroll (eds.): HCI Models, Theories and Frameworks: Toward a Multidisciplinary Science. Morgan Kaufmann (2003)
27. P. Chandler, J. Sweller. The Split-attention Effect as a Factor in the Design of Instruction. British Journal of Educational Psychology, Vol. 62. (1992) 233-246
28. J. Sweller, P. Chandler. Why Some Material is Difficult to Learn. Cognition and Instruction, Vol. 12. No. 3. (1994) 185-233