The Quality of Business Process Modelling Methods

Illustration of a Framework for Understanding Modelling Quality

Bart-Jan Hommes and Victor van Reijswoud

Bart-Jan Hommes: Delft University of Technology, Faculty of Information Technology and Systems, Department of Information Systems and Software Engineering, PO Box 356, 2600AJ, Delft, The Netherlands, email: hommes@is.twi.tudelft.nl

Victor van Reijswoud: Devote Institute of Technology, Devote nv, PO Box 73, 3980 CB Bunnik, The Netherlands, email : vreijswoud@devote.nl

Key words: Business process modelling, Quality, Evaluation of modelling methods

Abstract: The conceptual modelling of business processes is becoming popular. The number of methods and tools is growing fast. At the same time, an appropriate framework for understanding the quality of these modelling methods is lacking. In this paper we report upon the development of a framework for understanding the quality of business process modelling methods, called the Q-Me framework. The framework defines the elements that constitute a modelling method and presents a number of quality properties as well as ways to operationalise them. In this paper, the framework is illustrated by studying the quality of the Unified Modelling Language (UML) for the purpose of business process modelling. Conclusions are drawn both on the quality of UML and on the application of the framework to study UML.

1. INTRODUCTION

The modelling of business processes is becoming increasingly popular. Both experts in the field of Information and Communication Technology (ICT) and in the field of Business Engineering have come to the conclusion that successful systems (re)engineering starts with a thorough understanding of the business processes of an organisation: a business process model.

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35500-9_30

E. D. Falkenberg et al. (eds.), *Information System Concepts: An Integrated Discipline Emerging*

Conceptual modelling of business processes is deployed on a large scale to facilitate for instance Business Process Reengineering (BPR), ERP system implementation [Dav98], [Post96], Total Quality Management and Workflow Automation [Bar98].

The increasing popularity of business process modelling results in a rapid growing number of modelling methods. This increase in methods makes the process of selection and/or assembling a modelling method (e.g. method engineering [Har97]) more and more complex and time-consuming. Indicative for the huge range of different tools and methods is the overview of [Ket97]. An analysis of the Internet by the authors reveals approximately 350 business process modelling tools, all claiming to support 'effective', 'comprehensible', 'compact', 'suitable' etc. conceptual business modelling (see http://is.twi.tudelft.nl/~hommes/tools.html for this overview).

In fact, methods are seldom tested on these claims, since there is no framework available for assessing the quality of methods for conceptual business modelling. Existing frameworks for evaluating quality, focus on the quality of software- and information systems modelling methods, rather than on business process modelling methods. Apart from the fact that these frameworks do not have a business focus, there is criticism about the 'vagueness' of quality properties and the lack of operationalisation [Lind94], [Gil92].

The lack of appropriate means to assess (evaluate) the quality of this rapidly growing number of business modelling methods, and the dominant role these methods and tools can have in, for instance, Business Process Reengineering, ERP system implementation, Total Quality Management and Workflow Automation, justifies the development of a conceptual framework for understanding and evaluating the quality of these methods.

This paper reports on the first steps in developing a Quality-based Modelling Evaluation framework, called the Q-Me framework. The aim is to provide a set of well-defined quality properties and procedures to make an objective assessment of these properties possible. In section 2, a general framework for describing modelling methods will be presented. As an extension of this framework, quality properties for business modelling methods are identified in section 3. In section 4 the extended framework is illustrated by evaluating the quality of the Unified Modelling Language (UML) and its potential for business process modelling. After this, conclusions are drawn and directions for further research are presented in section 5.

2. MODELLING BUSINESS PROCESSES

Although the usage of modelling methods for understanding the information and business structure of organisations is increasing, the evaluation of these methods is a poorly developed scientific field. Some high-level frameworks have been proposed for the description of the elements in an information systems development methodology [Sel89], the evaluation and engineering of methods [Har97] and the application information systems methodologies in practice [Jay94]. However, these approaches do not focus in detail on the modelling methods used in a particular methodology. In this section we first introduce a general framework for describing modelling methods. In section 2.2 the area of business modelling methods is highlighted more specifically.

2.1 Framework

Based on the Method Theory as developed in [Gold98] and the general description of elements in an information systems methodology as described in the Framework for Understanding in [Sel89], we propose a framework that allows the description and evaluation of modelling methods. In line with [Sel89], we identify a *way of modelling* and a *way of working* in a modelling method. The Way of Modelling describes the models that are used in a method and the Way of Working describes the procedures by which these models are constructed. This division corresponds with the distinction between conceptual product method fragments and conceptual process method fragments as described in [Har97].

In the Way of Modelling we describe models by their constituting modelling concepts. The constituting modelling concepts are characterised by their notation and their meaning. We also describe the relationships between the different modelling concepts in one model and the notation of this relationship. The individual models are described by their mutual relationships as well as by their goals or purposes.

The Way of Working of a modelling method is described as a related set of activities together constituting the modelling procedure. Within the framework the procedure is specified at the model level. A complete overview of the way of working is achieved with the description of the mutual relationships between the procedures. Together the Way of Working and the Way of Modelling represent the perspective of the modelling method. By perspective we mean the view and purpose that guides the

conceptualisation of the real system into the conceptual system. The framework for evaluation is depicted in Figure 1.

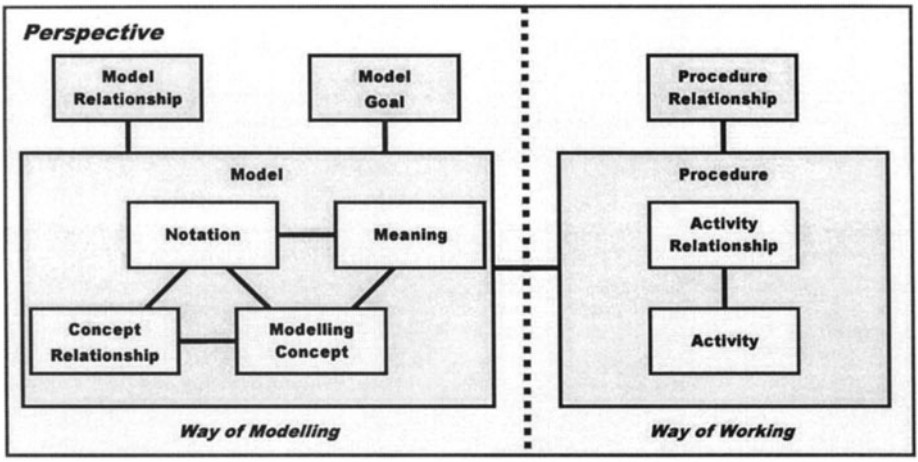


Figure 1. Framework for evaluation

For a detailed description of the four elements that constitute an individual model, we introduce a *modelling concept table*. In this table, each modelling concept is described together with its meaning and notation. The last column of the table describes the concept relationship by means of a meta model. The modelling language used to construct these meta models is similar to the ORM modelling language. A comprehensive overview of ORM (Object Role Modelling) can be found in [Hal98]. Other approaches, such as an ER approach [Chen77] or an object-oriented approach [Rum91] may also be used for meta modelling. As an example, the modelling concept table of a simple flowchart diagram is shown below.

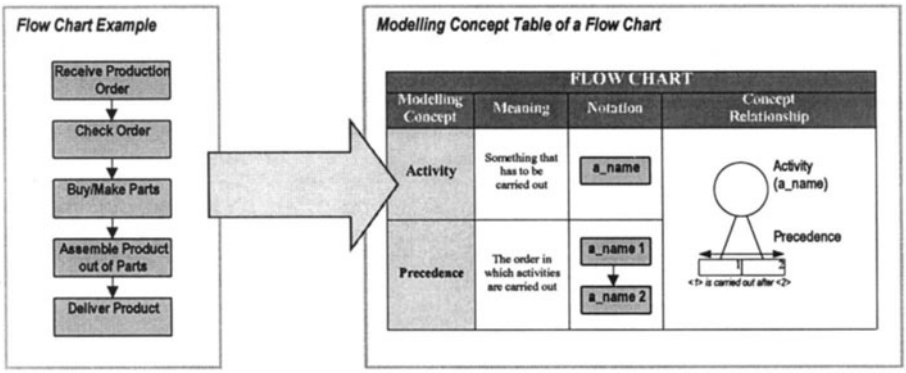


Figure 2. Simple Example of a Modelling Concept Table

A detailed description of the three elements that constitute the way of modelling is achieved by means of a *model table*. This table is drawn in the same fashion as the modelling concept table. It contains two columns specifying the models and their goals. The last column describes the model relationship by means of a meta model. The model table will be illustrated when the UML is described in section 4.

2.2 Modelling Business Processes

The interrelated set of modelling concepts that constitute the way of modelling represents an application domain. In the case of business processes modelling, this application domain is the business process. This section is a reflection of the existing consensus about what 'business processes' are and thus to what application domain the modelling concepts of the method should correspond. This is the *perspective* element of the framework presented in figure 1.

In the literature, a 'business process' is commonly defined as a chain of organisational or inter-organisational activities that are necessary to accomplish a product or service. Examples of this definition are "an ordering of work activities across time and place, with a beginning, an end, and clearly identified inputs and outputs" [Dav93] or "A set of activities that, taken together, produce a result of value to a customer" [Ham93]. We will refer to these definitions as definitions of a 'process' in general. The term 'business process' is reserved for a more specific class of processes.

Founded in [WiFl86], [Med92] and [Die94] we classify processes according to the nature of the activities that are carried out. If the nature of the activities is physical, such as assembling a product, then we speak of a *material process*. If their nature is about processing information, such as calculating the price of a product, then we speak of an *information process*. If the nature is about doing something with information, such as making a commitment to a supplier to pay for a product, then the corresponding process is called a *business process*.

Often 'core' and 'supportive' business processes are distinguished. A core (or primary) process is initiated from outside an organisation, e.g. the chain of activities that realises the delivery of a product to a customer. A supportive (or secondary) process is initiated from inside the organisation to provide support for core processes, e.g. buying new stock from a supplier.

A business modelling method should provide means to describe the dynamic aspects of the functioning of an organisation as well as the static characteristics of the information space on which the dynamic aspects build. Also the distinction between organisation and environment should be modelled, in order to distinguish between core and supportive processes.

3. THE QUALITY OF BUSINESS PROCESS MODELLING METHODS

This section addresses the quality of business process modelling methods. After a general definition of quality is given, quality properties specific to (business) modelling will be assigned to methods (section 3.1). Some attention to the operationalisation of these properties is paid in section 3.2.

3.1 Quality Properties of a Business Modelling Method

Quality has been defined in many ways, ranging from extremes as 'conformance to requirements' [Cro79] to 'fitness for use' [Jur79]. The International Standards Organisation (ISO) has attempted to unite the different views on quality in a general definition stating that quality is "the total of properties and characteristics of a product or service that are relevant for satisfying specific requirements and obvious necessities". This definition is taken as a starting point for a refinement of the definition of quality. Since the nuance between the meaning of the term 'property' and 'characteristic' is slight, we will not distinguish between them and use the term 'property' in the remainder of this paper. The 'product or service' under consideration is the business modelling method.

A common way to understand the quality of something is to subdivide quality in a number of quality properties that each address a particular aspect of quality. The evaluation of software quality by [Boe78] is an example of this approach. [Boe78] decomposes high level quality properties into lower level properties, resulting in a 'tree of quality properties'. This approach is adopted in this paper.

Three quality properties that provide a good basis for the evaluation of modelling methods are the properties particular to meta models of modelling languages as presented in the FRISCO report as formulated by the IFIP8.1 Working Group [Frisco96]. According to the FRISCO report, the following quality properties are important:

- *Expressiveness* - the degree to which a given modelling method is capable of denoting the models of any number and kinds of application domains;
- *Arbitrariness* - the degree of freedom one has when modelling one and the same domain;
- *Suitability* - the degree to which a given modelling method is specifically tailored for a specific kind of application domain.

The first two properties, viz. expressiveness and arbitrariness, are properties that are applicable for any modelling method, regardless of the domain that is modelled. The latter one, viz. suitability, is a property that is specific for the business process domain. Suitability for this business process domain is referred to as 'business suitability' [Bar98].

Both expressiveness and (business) suitability are quality properties of the way of modelling in particular. The properties are not orthogonal, they influence each other in such a way that an optimum has to be found [Frisco96]. A modelling method that is highly expressive contains modelling concepts that are generally applicable. Therefore, it has low business suitability. On the other hand, a modelling method that is highly suitable for business modelling contains concepts that are specific for the business domain. Therefore its expressiveness is low.

Arbitrariness is a property of the way of working in particular. Low arbitrariness limits the degree of freedom one has while modelling a domain. A low degree of freedom during the modelling process results in a way of working with results that are reproducible. Since particularly low arbitrariness is a quality of a way of working, we will use the opposite term '*determinism*' in stead of arbitrariness in the remainder of this article.

Due to the fact that the three properties that were mentioned above specifically address the meta model of the modelling language, their contribution to the overall quality of a modelling method is restricted to the *modelling concept*, *meaning* and *concept relationship* elements of the framework. In order to cover all the elements that constitute a method, other properties are necessary. Other properties that are proposed in literature (e.g. [Bar98], [Har97]) are:

- *Comprehensibility* - the ease with which the way of working and way of modelling are understood by the participants;

- *Coherence* - the degree to which the individual sub models of a way of modelling constitute a whole;
- *Completeness* - the degree to which all necessary concepts of the application domain are represented in the way of modelling;
- *Efficiency* - the degree to which the modelling process utilises resources such as time and people;
- *Effectiveness* - the degree to which the modelling process achieves its goal.

The properties are summarised in figure 3. First of all, product quality and process quality are distinguished. They refer to the quality of respectively the way of modelling and the way of working of a modelling method. For each property, its area of application and the knowledge that is required to study the property is presented. The figure is a reflection of the informal introduction of the properties in the previous paragraphs, combined with the framework for modelling methods as presented in section 2.1. Note that 'arbitrariness', as mentioned in [Frisco96] has been replaced with its opposite 'determinism'.

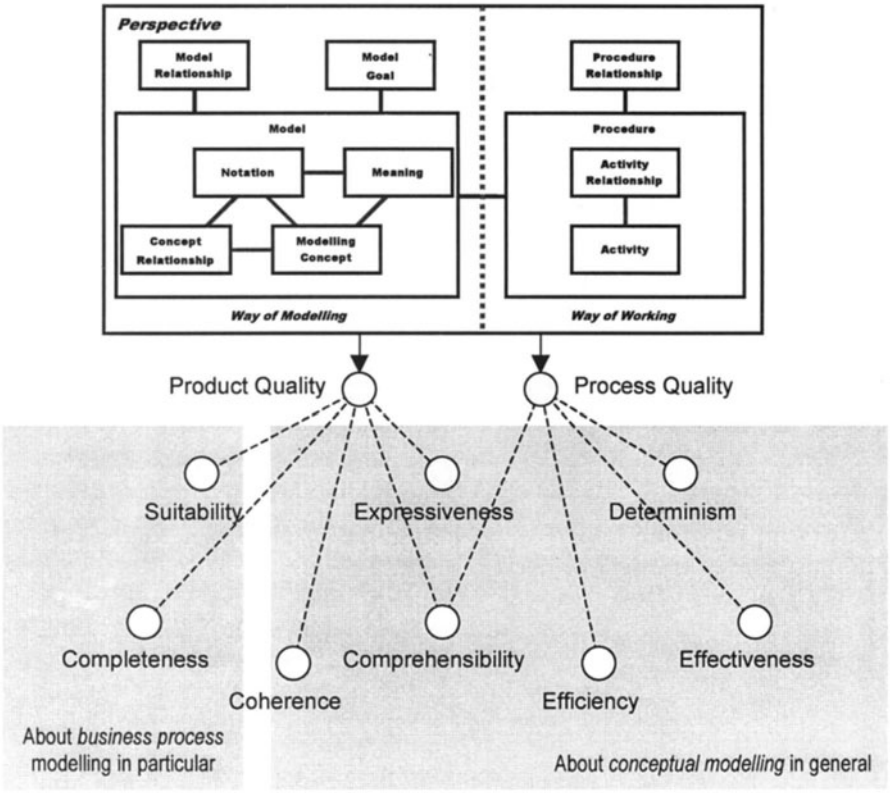


Figure 3. The Quality of a Business Modelling Method

3.2 Measuring the Quality Properties

In this section we discuss the operationalisation of the quality properties that were presented in the previous section. We will discuss how the quality properties relate to the elements of the framework that was introduced in section 2.1 and how the proposed *modelling concept table* and *model table* contribute as means for the measurement of the quality properties.

Suitability, Completeness - The operationalisation of the properties that have a business focus, viz. suitability and completeness, comprehends a comparison between the modelling concepts of the method that is evaluated and the consensus that there is about the concepts that should comprise the business domain, as was discussed in section 2.2. As such, the elements *modelling concept* and *meaning* of the framework provide the basis for doing this. This shows for instance that the flow chart example of figure 2 is *suitable*, since it is able to express a process as activities and some form of interrelation between them. It is however *incomplete*, since it lacks the appropriate concepts for distinguishing between core and supportive processes, material-, information- and business processes, etc.

Coherence - The two elements in the framework that provide a basis for the operationalisation of the coherence property are the elements that refer to the structure of the way of modelling, viz. the *concept relationship* and the *model relationship* elements. The concept relationship is described by means of a meta model in the last column of the modelling concept table. A model is coherent when there are no isolated parts in its meta model. The overall coherence of a way of modelling is described by a meta model that combines the meta models of the individual models. The way of modelling is coherent when this overall meta model does not contain isolated parts. Furthermore, a meta model of the way of modelling is the basis for an understanding of how the individual models constitute a coherent whole.

Expressiveness - Expressiveness is the degree to which a modelling method is capable of modelling any number and kinds of application domains. The specificity / generality of the *meaning* of *modelling concepts* is an indication for the level of expressiveness. In literature, meta model transformations are mentioned as a measure for expressiveness [Frisco96]. When there is a mapping from a meta model of a method A to the meta model of a method B without loss of meaning, it can be derived that B is at least as expressive as A. Furthermore, the ability of a method to describe its own concepts is an indication for high expressive power.

Comprehensibility - The elements *meaning*, *notation* and *modelling concept* provide a useful 'triangle' for an operationalisation of the comprehensibility property of a modelling method. Consistency between the notation-meaning and meaning-modelling concept relations is a measure for comprehensibility. The flow chart example is comprehensible in that sense. The notation-meaning relation of the 'flow' concept is consistent because an arrow is a perfect means for sequencing activities. It is obvious that omitting the arrowhead from the arrow notation would make the model very incomprehensible.

Another way of operationalising comprehensibility is to consider the amount of different modelling concepts per model. The lower the number of modelling concepts, the easier the model is to comprehend. This also holds for the deviation in the number of concepts per model.

Determinism - The degree of freedom one has when modelling one and the same domain using a method is reflected in the determinism property. When there is only one way of modelling a domain using a method, the method is said to be *deterministic* or has zero arbitrariness. Arbitrariness is introduced in a method when there are different modelling concepts or structures of modelling concepts in a method that have a same meaning. If this is the case, different models can model the same domain and freedom is introduced. The *meaning*, *modelling concept* and *concept relationship* need to be evaluated to determine the determinism of a method.

Quality Property \ Framework Element	Modelling Concept	Concept Relationship	Notation	Meaning	Model Relationship	Model Goal	Procedure	Procedure Relationship	Activity	Activity Relationship
Suitability	■			■						
Completeness	■			■						
Coherence		■			■					
Expressiveness	■	■		■						
Comprehensibility	■		■	■						
Efficiency							■	■	■	■
Determinism	■	■		■						
Effectiveness						■	■	■	■	■

Figure 4. Quality properties and requested elements for evaluation

Effectiveness, Efficiency - Effectiveness on the level of a single model is the extent to which the goal of that model is achieved by means of the interrelated set of activities that constitute the procedure for that model. It is clear that the elements *model goal*, *activity* and *activity relationship* of the framework need to be evaluated in order to operationalise the effectiveness of a method. Evaluation of the effectiveness of the whole method also requires an assessment of the *procedure relationship* and *model relationship*.

In this paper, no attention is paid to the evaluation of effectiveness and efficiency of a method. This is due to the fact that not much is know yet about how to evaluate the framework elements that refer to the way of working of the method. This is not problematic for the evaluation of the UML in the next section. Since it is a modelling language, the emphasis is on the way of modelling rather than the way of working.

Figure 4 shows which elements of the framework need to be evaluated in order to operationalise the quality properties of the framework.

4. THE QUALITY OF THE UNIFIED MODELLING LANGUAGE

In this section, the proposed framework is illustrated by applying it to draw conclusions on the quality of the Unified Modelling Language (UML) for the purpose of business process modelling. Most of the description of UML is based on [OMG99] and [Fow97]. The aim of this section is to illustrate the application of the Q-Me framework.

4.1 UML put into the Framework

In this section, the UML is described using the Q-Me framework that was introduced in section 2.1. Since the UML is a modelling language rather than a method [Fow97], the focus will be on the way of modelling of the framework. First of all, the *model* and *model goal* elements of the framework are presented by means of a model table (table 1). This table describes the most prominent models of the UML and their goals.

WAY OF MODELLING	
Model	**Model Goal**
Use Case	Modelling the interactions between the system that is to be modelled an its environment by means of environmental actors and their use of the system (use cases)
Class	Modelling the static structure of the systems object- and subject world by means of a classification of objects and their interrelationships
Sequence	Modelling the way that the objects interact within one use case
State Transition	Modelling the internal states and transitions of a single object class
Activity	Modelling the overall system behaviour by means of an interrelated set of tasks

Table 1. Model Table of the UML

The framework elements that constitute a model, viz. modelling concept, notation, meaning and concept relationship are set out in detail by means of modelling concept tables. Due to the limited space in this paper, we present the modelling concept tables of three models: the use case-, class- and sequence diagrams, since these models are most often used.

The construction of a *Use Case Model* is often the first step in a project that uses UML as a notation. It is a model that describes system interactions, i.e. interactions between a computer system and its environment. Two modelling concepts are characteristic for the use case model: the concept of an 'actor' and the concept of a 'use case'. According to [Fow97], an 'actor' is a role that a user plays with respect to the system. Actors carry out 'use cases' which can be seen as pieces of user functionality of the system. Furthermore, it is possible to model relations between individual use cases, so called 'extends' and 'uses'. The procedure for identifying actors and use cases is not clear. Fowler describes four alternatives for doing this [Fow97, p.46]. An overview of the use case model is presented in table 2.

The class model is considered to be the core of any object oriented modelling method and as such also for the Unified Modelling Language. In this paper the most important modelling concepts of the class diagram are discussed. An elaborate discussion can be found in [OMG99]. The class model describes the types of objects that are part of the system and their mutual relationships. In order to maintain clarity later on, the table below (table 3) shows only the most important modelling concepts of this model.

USE CASE MODEL			
Modelling Concept	**Meaning**	**Notation**	**Concept Relationship**
Use Case	A coherent unit of functionality provided by a system	uc_name	
Actor	An interactor in the system environment	ac_name	
UC_ Association(*)	The participation of an actor in a use case	ac_name, uc_name	
Extension	One use case extending the functionality of another	uc_name 1, «extend», uc_name 2	
UC_ Generalisation(*)	One use case being offering more specific functionality than another	uc_name 1, uc_name 2	
Inclusion	The (re)use of overlapping functionality	uc_name 1, «include», uc_name 2	

(*) In the OMG standard UML V1.3, the names of modelling concepts are unique within a single aspect model, not within the whole model cycle.' Association' for example, is a modelling concept used in the Use Case Model as well as in the Class Model. We prefer to give modelling concepts unique names within the whole model cycle and added prefixes to conflicting concept names. In general, the names of the modelling concepts correspond to the names mentioned in the OMG standard UML V1.3. However, in the OMG standard. This remark hold also for the modelling concept tables in the remainder of this article.

Table 2. Modelling Concept Table of the Use Case Model

CLASS MODEL			
Modelling Concept	**Meaning**	**Notation**	**Concept Relationship**
Class	A descriptor of a set of objects with similar structure, behaviour and relationships	cl_name	
Attribute	A property of a class	cl_name, at_name 1, at_name n	
Operation	A service that an instance of the class may be requested to perform	cl_name, op_name 1, op_name n	
CL_Association	A static n-ary relationship between classes	Binary: cl_name, cl_name; n>2-ary: cl_name, cl_name, cl_name	
Association End	Identification of a class in an association	cl_name	
Composition	A form of aggregation with strong ownership and coincident lifetime of the part (2) with the whole (1)	cl_name 1, cl_name 2	
CL_ Generalisation	Relationship between a generic class (1) and a more specific class (2) with additional behaviour and structure	cl_name 1, cl_name 2	

Table 3. Modelling Concept Table of the Class Model

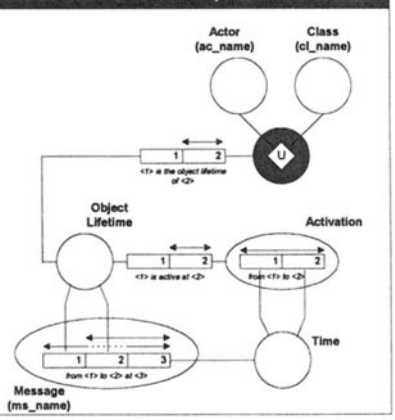
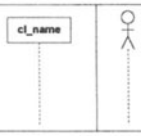
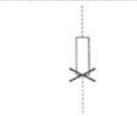
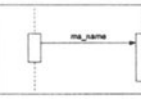
SEQUENCE MODEL			
Modelling Concept	Meaning	Notation	Concept Relationship
Time	A point in time	Represented by the Y-axis in the diagram	
Object Lifetime	The instantiation of a class or actor in a specific role		
Activation	A period in time during which an instantiation of a class or actor if performing action		
Message	Specification of a stimulus from a sending to a receiving activation at a point in time		

Table 4. Modelling Concept Table of the Sequence Model

A sequence model describes the collaboration between object classes over time. It assumes that objects collaborate by sending messages to each other in a sequence. Messages are represented by horizontal arrows. The time dependency between messages is expressed by the vertical ordering of the horizontal arrows in the diagram. An overview of the main concepts is presented below (table 4).

The description of the *model relationship* element of the evaluation framework encompasses the integration of the individual meta models as presented in the concept relationship column of the modelling concept tables (table 2, 3, 4). This integration results in an overall meta model which is presented in figure 5.

The dotted ellipses in this diagram are an extension to the ORM notation as described by [Hal98]. They can be compared to what is referred to as a 'schema type' in [Hof93]. They group the modelling concepts that belong to one single sub model. The concept itself is a specialisation of the ORM object type concept, identified by its population. Since it is a specialisation of an object type it can participate as a role in new fact types, of which the 'UC/SQ Interrelationship' fact type is an example. It expresses the interrelationship between sequence model and use case model: within UML, a sequence model is drawn for each use case.

The meta model shows that there is no direct interrelationship between the class model and the use case model. The relationship between these models is established by the sequence model. For every use case, a sequence

model can be drawn. This sequence model addresses the classes of the class model by means of 'object lifetimes'.

Furthermore, the meta model shows that there is a double interrelationship between the class model and sequence model. On the one hand, the 'class' of the class model is related to the 'object lifetime' in the sequence model. On the other hand, operations on classes are related to messages in the sequence model.

This same double relationship holds for the use case model and the sequence model. On the one hand, a sequence model can be drawn for each use case. On the other hand, the 'actor' concept of the use case model is related to an 'object lifetime' in the sequence model.

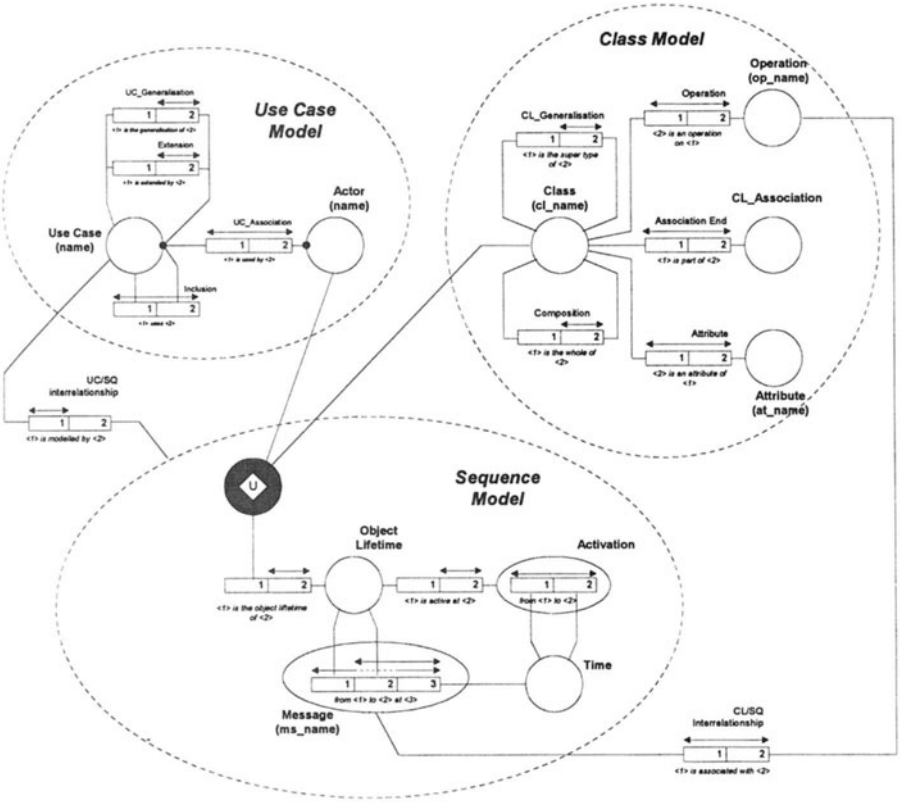


Figure 5. Model Relationship

The activity- and state transition model are not discussed in detail in this paper. When they would have been placed in the meta model it would become clear that there is a strong relation between the state transition model

and the class model, since each state transition model describes the states and transitions of one class. Furthermore it would become clear that the activity model is isolated from the other models, since no relationships could be found.

4.2 Business Modelling and UML

Recently, the application of UML for modelling business systems has gained popularity [Pen99], [Hru98], [UML97]. In [UML97], the standard UML extension mechanisms are used to introduce modelling concepts that are more business suitable. The same approach is adopted in [Hru98] to introduce workflow suitable modelling concepts. Since these mechanisms extend the meta model with additional sub types to the existing object types, they do not have a large impact on the general UML that is presented in this paper. The authors do not know the approach adopted in [Pen99], since it was not published during the writing of this paper.

4.3 Conclusions on the Quality of UML

In this section, conclusions with respect to the quality of UML are drawn. In order to do so we will pass the quality properties that were mentioned in section 3.2 in review. For each property, the conclusions are underpinned by the evaluation of the framework elements that are applicable for that property.

Suitability, Completeness - An evaluation of the modelling concepts and their meaning reveals that the UML is a very general approach which is not exclusively suitable for business process modelling. The modelling concepts in the business domain (see section 2.2) are not easily mapped to UML concepts. Some business suitability can be found in use case model and the activity model, with which it is possible to model processes as described by [Dav93] as "an ordering of work activities across time and place". Since it does not for instance distinguish between material-, informational- and business activities, core- and supportive processes, it can be considered *incomplete* with respect to the UML concepts covering the business domain.

Coherence - An evaluation of the individual UML models reveals that the concept relationship of each model forms a coherent whole, i.e. that their corresponding meta models do not show any isolated parts.

The interrelationship between the aspect models of UML is less coherent. There is no direct relationship between the use case model and the class model, which means that it is not possible to keep these models consistent without using sequence models in a modelling project. The double relationship between class- and sequence model as well as the double relationship between sequence- and use case model makes it unnecessarily difficult to maintain consistency.

The activity diagram, which was briefly introduced in table 1, but has not been presented in detail due to the limited space in this paper, does not have any coherence with the rest of the UML models. This is especially regrettable since it is a model with some business suitability as has been concluded before.

Expressiveness - The generality of the meaning of the modelling concepts reveals that UML has a high expressive power. This conclusion is in line with the conclusion on the low business suitability, taking into account the dependency between both properties as discussed in section 3.1. Especially the class model, which is to be considered as the core of any object oriented method is build up out of modelling concepts with a very general meaning. Illustrative is the concept of a 'class'. Another indication of high expressive power is the fact that the meta model of UML can be described in its own class model, see for instance the formal semantics of UML in [UML97].

Comprehensibility - It can be concluded that UML is a comprehensible modelling method. Meaning, notation and modelling concept form a consistent whole. For instance, the notation of the concept 'class' is a rectangle, both in the class diagram and in the sequence model. Arrows represent concepts that deal with 'flows' of something, except for the arrows between use cases in the use case diagram. Another indication for comprehensibility is the number of concepts per model. We can conclude that this number is well balanced over all models.

Determinism - With respect to determinism it can be concluded that there are some weak points with respect to the degree of freedom one has when modelling one domain. An unwanted degree of freedom is present in the class model in which the meaning of the concept 'association' and 'attribute' overlaps (see [Fow97, p. 63]). A 'Colour' is, for instance, a property of a 'Car' but might as well be a descriptor of a group of things that have same properties (e.g. all colours have the property to be perceived by the eye). The difference between 'association' and 'class' is also not clear. 'Marriage' is for

instance 'a binary relationship between two persons but also a descriptor for a group of concepts with similar properties (e.g. that they can be happy or unhappy). The result is that several class models can be drawn that represent the same domain correctly, which means that the results of the modelling process are not reproducible.

Furthermore, a consensus about the meaning of the concept 'UC_Association' in the use case diagram is lacking. In [Fow97, p.46] four alternative meanings can be found. This does also not contribute a deterministic modelling process with reproducible results.

Effectiveness, Efficiency - As has been mentioned before, the *effectiveness* and *efficiency* properties are not evaluated here because not much is known about the way of working in UML. This is due to the fact that UML is a language rather than a method.

5. CONCLUSIONS ON THE APPLICATION OF THE FRAMEWORK

In this paper we have proposed and illustrated the Q-Me framework for the evaluation of the quality of business modelling methods. The framework allows an assessment of both the product quality and the process quality of modelling methods, with regard to a set of properties that have been defined in the literature. The increasing usage of business modelling methods for reengineering, ERP implementation, TQM and workflow automation projects requires a framework which allows analysts and users to assess the quality of the vastly increasing amount of available methods.

The application of the Q-Me framework on the UML revealed both some strengths and weaknesses. The most important benefit of the framework is that it provides a set of categories and properties that allows a uniform and formal description of the model elements within one model type as well as the different model types used within one modelling method.

A uniform characterisation of a modelling method forms a precondition for comparison. The meta modelling method used in the Q-Me framework enables a good evaluation of coherence between the modelling concepts and models used in a modelling method. This property is important when designing support for the method. Also, the level of comprehensibility is revealed by the meta modelling method. Large numbers of concepts and

relationships between the concepts decrease the ease with which users and analysts master a method.

A major shortcoming of the current status of the Q-Me framework is the lack of a formal definition of the operationalisation of properties and the absence of a quantifiable metric to express the quality of a business modelling method. Although the framework allows a characterisation on the basis of individual properties, the lack of an overall metric makes it difficult to make an objective comparison on the quality of different methods.

Finally, the application of the framework to the UML failed to test the effectiveness and efficiency since the UML has no pre-defined modelling process.

Future research will be conducted to improve the framework. The operationalisation of the introduced quality properties will be studied in more detail. Especially properties that relate to the way of working, such as effectiveness and efficiency will get attention. Furthermore, the theory will be validated by means of the application of the framework to other business process modelling methods. The results of this application will be compared with the opinion of experts in the field of business process modelling.

6. REFERENCES

[Bar98] Barros, A. P., Hofstede A. H. M. ter, *Towards the Construction of Workflow Suitable Conceptual Modelling Methods,* Information Systems Journal, October 1998;

[Boe78] Boehm, B., *Characteristics of Software Quality*, North-Holland, New York, 1978;

[Chen77] Chen, P. P., *The Entity-Relationship Model: towards a Unified View of Data*, ACM-TODS, 1977;

[Cro79] Crosby, P. B., *Quality is Free*, McGraw-Hill, London, 1979;

[Dav93] Davenport, T. H., *Process Innovation: Reengineering Work Through Information Technology,* Harvard Business School Press, Boston, 1993;

[Dav98] Davenport, T. H., *Putting the Enterprise into the Enterprise System*, Harvard Business Review, July-August 1998;

[Die94] Dietz, J. L. G., *Business Modelling for Business Redesign*, In: Proceedings of the 27th international conference on system sciences, IEEE Press, Los Alamitos, August 1994, pp. 723-732;

[Fow97] Fowler, M., Scott, K., *UML Distilled*, Addison-Wesley, Boston, 1997;

[Frisco96] Falkenberg, E. D. *et al, A Framework of Information System Concepts* (IFIP WG 8.1 Task Group FRISCO), Leiden University, Leiden, 1996;

[Gil92] Gillies, A. C., *Software Quality,* Chapman & Hall computing, London, 1992;

[Gold98] Goldkuhl, G., Lind, M., Seigerroth, U., *Method Integration as a Learning Process*, in: Jayaratna, N., Fitzgerald, B., Wood-Harper, T., Larasquet, J-M (Eds.), *Training and Education of Methodology Practitioners and Researchers*, Springer-Verlag, London;

[Hal98] Halpin, T., *Object Role Modeling (ORM/NIAM)* in: Handbook on Information Systems Architectures, Springer-Verlag, Heidelberg, 1998;

[Ham93] Hammer M., Champy, C., *Reengineering the Corporation: A Manifesto for Business Revolution*, Brealey, London, 1993;

[Har97] Harmsen A. F., *Situational Method Engineering*, Ph.D thesis, Moret Ernst & Young Management Consultants, Utrecht, 1997;

[Hof93] Hofstede, A. H. M. ter, *Information Modelling in Data Intensive Domains*, Ph.D thesis, University of Nijmegen, Nijmegen, 1993;

[Hru98] Hruby, P., *Structuring Specification of Business Systems with UML (with an Emphasis on Workflow Management Systems)*, Springer-Verlag, Heidelberg, 1998;

[Jay94] Jayaratna, N., *Understanding and Evaluating Methodologies; NIMSAD, a Systemic Framework.* McGraw-Hill, London, 1994;

[Jur79] Juran, J. M., *Quality Control Handbook,* McGraw-Hill, London, 1979;

[Ket97] Kettinger, W.J., *Business Process Change: A Study of Methodologies, Methods, and Tools*, MIS Quarterly, March 1997;

[Lind94] Lindland, O.I., Sindre, G., Sølvberg, A., *Understanding Quality in Conceptual Modelling*, IEEE Software, March 1994;

[Med92] Medina-Mora, R., Winograd, T., Flores, R., Flores, F., *The Action Workflow Approach to Workflow Management Technology,* CSCW'92 Proceedings, ACM, New-York, 1992;

[OMG99] Rational, UML Standard Version 1.3 Alpha R5, www.rational.com/uml, March 1999;

[Pen99] Penker M., Eriksson, H. J., *Business Modelling With UML; Business Patterns and Business Objects*, John Wiley & Sons, New-York, to be published 1999;

[Post96] Post H. A., Es, R. M. van (Eds), *Dynamic Enterprise Modeling: A paradigm Shift in Software Implementation,* Kluwer, Dordrecht, 1996;

[Rum91] Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., Lorensen, W., *Object-Oriented Modeling and Design*, Prentice-Hall, Englewood Cliffs, 1991;

[Sel89] Seligmann, P. S., Wijers, G. M., Sol, H. G., *Analysing the Structure of IS methodologies, an Alternative Approach*, in: Proceedings of the 1st dutch conference on information systems, Amersfoort, 1989;

[UML97] Rational, *UML Extension for Business Modeling* version 1.1, www.rational.com/uml, 1997;