Graphical modeling techniques and usefulness in the Model Driven Arcitechture: Which are the criteria for a “good” Computer independent model?

Lars-Olof Johansson, Magnus Wärja, Harald Kjellin and Sven Carlsson

Kristianstads University, Lunds University, Kristianstads University and Lunds University

lars-olof.johansson@hkr.se, magnus.warja@ics.lu.se, harald.kjellin@hkr.se and sven.carlsson@ics.lu.se

Abstract. The Object Management Group (OMG) has presented a framework for model driven development: Model Driven Architecture (MDA). The departure point of this paper is an interest in process oriented modeling techniques relevant for a computer independent model in the MDA. We searched for the answer on the question what constitutes a “good” computer independent model (CIM). To get an answer we were using OMG:s demands of the modeling techniques that should be used when developing CIMs and found four process oriented modeling techniques which could serve as candidates. These four process oriented modeling techniques is analyzed based by Moody’s quality criterions. The findings indicates that there are only one candidate, BPMN, that meets some of Moody’s quality criterions, but there also drawbacks in BPMN, such as the criterions; expressiveness and emphasis.

1. Introduction

In 2003 the Object Management Group (OMG) presented a framework for model driven architecture (MDA) which has attracted a lot of attention in the field of software engineering and information systems development (ISD) (Miller and Mukerji 2003). The basic idea behind MDA is to be model-driven because it provides a means for using models to direct the course of understanding, design, construction, deployment, operation, maintenance and modification [1]. In other words the primary focus should be on software development models rather than on programming code (Selic 2003).

The MDA framework consists of models at four different levels: computer independent models (CIM), platform independent models (PIM), platform specific models (PSM) and code. Since the presentation of the MDA-framework researchers and practitioners have showed a great interest in PIM, PSM and code but lesser interest in CIM (Johansson, Cronquist et al. 2007; Johansson, Cronquist et al. 2007; Wärja 2007). Much of the literature has a specific focus on transformations between PSM and code.

The departure point of this paper is an interest in graphical modeling techniques relevant for a CIM (Miller and Mukerji 2003) and a search for the answer to the question what constitutes a "good" computer independent model. One major reason behind this interest in CIM is that its seems like the research community (and practitioners) is focusing on the more technical aspects of MDA instead of focusing on how should CIM support the understanding of a problematic and complex situation in an organization.

The research question addressed is: Which are the relevant graphical modeling techniques to be used when creating computer independent models in the MDA framework? Hence, we will: 1) identify relevant graphical modeling techniques for CIM, 2) identify relevant quality criterions for evaluation of graphical modeling techniques for CIM, and 3) evaluate graphical modeling techniques for CIM using the identified quality criterions.

One of the main motives behind the research question and aim is that graphical diagrams are believed to be more effective than text in communication between end users and/or domain practitioners (Moody 2006).

The research started with a literature review with the aim to identify relevant graphical modeling techniques that could useful when producing a CIM. We started by reviewing well known articles about graphical modeling techniques and identified references in order to distinguish the most prominent graphical modeling techniques that could be used for a CIM in MDA. Since most models used for MDA have a process perspective we choose to exclude non process oriented graphical modeling techniques. We found four candidates: 1) **Flowchart/node maps, 2) Business Process Modeling Notation (BPMN), 3) Event Driven Process Chains (EPC), and 4) UML-activity diagrams**. These four different modeling techniques will be discussed from the viewpoint what constitutes a "good" modeling technique for developing a CIM based on the theoretical frameworks.

2. Quality criterions and computer independent models in MDA

In this section we will start with an overview of MDA (Section 2.1) and thereafter we will present quality criterions from two sources: Object Management Group (Section 2.2) and Daniel Moody (Section 2.3).

2.1 Overview of MDA

We will describe the basic ideas and definitions of Model Driven Architecture (MDA) based on how it is presented by the Object Management Group [1]. The basic idea behind MDA is to be model-driven because it provides a means for using models to direct the course of understanding, design, construction, deployment, operation, maintenance and modification (Miller and Mukerji 2003).

OMG defines model as: “A model of a system is a description or specification of that system and its environment for some certain purpose. A model is often presented as a combination of drawings and text” (Miller and Mukerji 2003) and system as “system may include anything: a program, a single computer system, some combination of parts of different systems, a federation of systems, each under separate control, people, an enterprise, a federation of enterprises…Much of the discussion focuses on software within the system” (Miller and Mukerji 2003). MDA is using graphical models on three different levels: CIM, PIM and PSM. Each level is using models at a certain abstraction level. The different levels of models in MDA is mapped/transformed to the underlying level, a CIM can be traceable to a PIM by a simple transformation and a PIM can be translated to a PSM, and the PSM can then be translated to code (Figure 1).

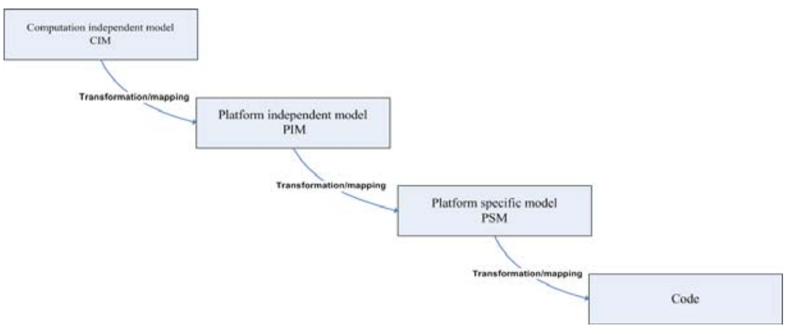


Figure 1: Overview of MDA

In the MDA-framework the CIM is at the “highest” level, placed in the top-left corner in Figure 1 (Miller and Mukerji 2003).

The **CIM** plays an important role in bridging the gap between domain experts with their requirements and experts of the design and construction of suitable artifacts (Miller and Mukerji 2003).

The **PIM** focuses on the operation of a system while hiding the details necessary for a particular platform. PIM exhibits a specified degree of platform independence so as to be suitable for use with a number of different platforms of similar type (Miller and Mukerji 2003).

The **PSM** combines the platform independent viewpoint with an additional focus on the detail of the use of a specific platform by a system. A PSM combines the specifications in the PIM with the details that specify how that system uses a particular type of platform (Miller and Mukerji 2003).

Model **transformation** is the process of converting for a specific system one model to another model. The PIM and other information are combined by the transformation to produce a platform specific model (PSM). An MDA mapping provides specifications for a transformation of a PIM into a PSM for a particular platform (Miller and Mukerji 2003).

This paper will focus CIM and graphical modeling techniques that supports the making of "good" CIM.

2.2 OMG's quality criterions

In OMG's MDA-framework there are at least seven demands/requirements that has to be fullfilled for a chosen graphical modeling technique that is to be used in the CIM (Miller and Mukerji 2003):

1. The modeling technique should aim at describing the environment (people and organizations) of the computerized system
2. The modeling technique should aim at describing the situation in which the system will be used
3. The modeling technique should be useful when eliciting the requirements for the system and the requirements should be traceable to the PIM
4. The primary user of the modeling technique should be the domain practitioner.
5. The modeling technique should not describe the functionality of the computerized system
6. The model should be data independent (a CIM does not show the details of the system's structure)
7. The model should bridge the gap between domain experts with their requirements and experts of the design and construction of suitable artifacts.

2.3 Moody’s quality criterions

Graphical diagrams are believed to be more effective then text in the communication between end-users and/or domain practitioners (Moody 2006). Communication effectiveness can be measured by speed, ease, and accuracy in which the information can be understood. According to Moody the design of graphical models in information systems development (ISD) has been based on personal taste, intuition, consensus rather than scientific theory and evidence (Moody 2006). The same graphical symbols – variants of boxes and arrows – are used over and over again while other attributes such as colors, size, spatial layout and value are seldom used. Many graphical notations in ISD are not consistent with principles of graphical design (visual appearance or form). The importance of diagrams have been grossly understated by ISD researchers. For example, the details of a graphical syntax (how to visually represent these constructs) are treated as being of little or no consequence. Even slight changes in graphical representation can have dramatic impacts on understanding and problem solving performance. This suggests that the form of diagrams is just as important – if not more – than their content.

Moody (2006) presents nine principles for graphical modeling: discriminability, perceptual and cognitive limits, emphasis, cognitive integration, perceptual directness, structure, identification, expressiveness, and simplicity. Moody does not use these nine principles when discussing different notations of graphical modeling techniques instead he uses the principles to discuss benefits and drawbacks of a specific model (created with the help a modeling technique).

Discriminability means easy to see and to differentiate from one another. It discusses the elements of a diagram. Moody separates between absolute and relative discriminability:

- Absolute discriminability: the ability to see diagram elements and separate them from the background. It is determined by three primary factors size, contrast and proximity.
- Relative discriminability: the ability to differentiate between different types of diagram elements. It is determined by the number and size of differences between symbols used to represent different constructs. The greater the perceptual variation between symbols used to represent different constructs, the faster and more accurately they will be recognized.

Human beings have both **perceptual and cognitive limits**. One of the most common mistakes in ISD diagramming practice is to show too much information on a single diagram. This results in a diagram that acts as a barrier instead of a communicative aid.

- Perceptual limits: the ability to visually discriminate between diagram elements decreases with their number and proximity. In general, the difficulty of discerning diagram elements increases quadratically with diagram size.

- Cognitive limits: the number of diagram elements that can be comprehended at a specific moment is limited by working memory capacity, which is believed to be “seven plus or minus two” concepts at a specific moment. When this is exceeded, a state of cognitive overload ensues and comprehension degrades. One of the most effective ways of reducing complexity is to divide them into smaller parts, decomposition or modularization.

In most ISD diagrams, all elements look the same: there is no way of telling which are the most important. The most important concepts should be **emphasized** (highlighted) to bring them to the readers’ attention, while less important or background elements should be de-emphasized (lowlighted).

It is very common in ISD diagramming to use multiple diagrams. Many developers use for instance UML which consists of 13 different modeling notations. The notation should provide an explicit mechanism to **support cognitive integration**:

- Conceptual integration: enabling the reader to integrate information from separate diagrams into a coherent mental representation of the problem.

- Perceptual integration: providing perceptual cues (orienting, contextual and directional information) to aid navigation between diagrams.

Perceptually direct representations are representations whose interpretation is spontaneous or natural, in that their meaning can be extracted automatically by the perceptual system.

- Representation of constructs: icons are symbols which perceptually resemble the objects they represent. Using icons to represent constructs reduces cognitive load.

- Representation of relationships: perceptual directness also applies to representation of relationships among diagram elements. Certain spatial configurations of diagram elements predispose people towards a particular interpretation even before the meaning of the elements is understood.

Organizing diagram elements by a **structure** into perceptual groups expands the number of elements that can be shown on each diagram without exceeding cognitive limits.

Moody divides the concept of **identification** into external identification and internal identification. External identification defines the correspondence between the diagram and the represented world. Diagram elements (both nodes and links) should be clearly labeled, using terminology familiar to domain experts to help trigger domain knowledge. Internal identification defines the correspondence between graphical conventions and their meaning. A key should be used summarizing the graphical convention.

Most graphical modeling techniques in ISD use a very limited graphical vocabulary (boxes and arrows) which implies a low **expressiveness**. We could use the full range of visual variables such as shape, colors, size, value, orientation and texture.

The number of graphical conventions used in a notation should be limited (use visual variables instead). Keep it simple, let **simplicity** rule.

3. Matching CIM with modeling techniques

This section aims at matching CIM with graphical modeling techniques presented in the ISD-literature. The proposed modeling techniques are characterized as process oriented modeling techniques. The motive for the choice of the four process modeling techniques is that they are widely used in describing problematic situations in organizations.

Event-driven process chains (EPC), business process modeling notation (BPMN) (Recker 2006; Recker, Indulska et al. 2006), UML-activity diagrams (Ambler 2004; Ambler 2007) and flowcharts are all examples of process oriented modeling techniques. In process-oriented modeling techniques there is always a starting point (input) and an end point (output) (Mathiassen, Munk-Madsen et al. 2000; Vidgen, Avison et al. 2002a). Petri nets and YAWL (Yet Another Workflow Language) are also examples of graphical modeling techniques that have gained a lot attention in the research community. These two graphical modeling techniques are mostly used to create PIM not CIM and are therefore not discussed in this paper. The four proposed graphical modeling techniques are commonly used by both practitioners and researchers when it comes to modeling and describing an environment (people and organizations) of a computerized system.

There are four examples of graphical process oriented modeling techniques **flowchart/node-maps**, **EPC, BPMN,** and **UML-activity diagram** connected to CIM. (We assume that flowcharts are well known and do not need a further description.) These four techniques share the same problems:

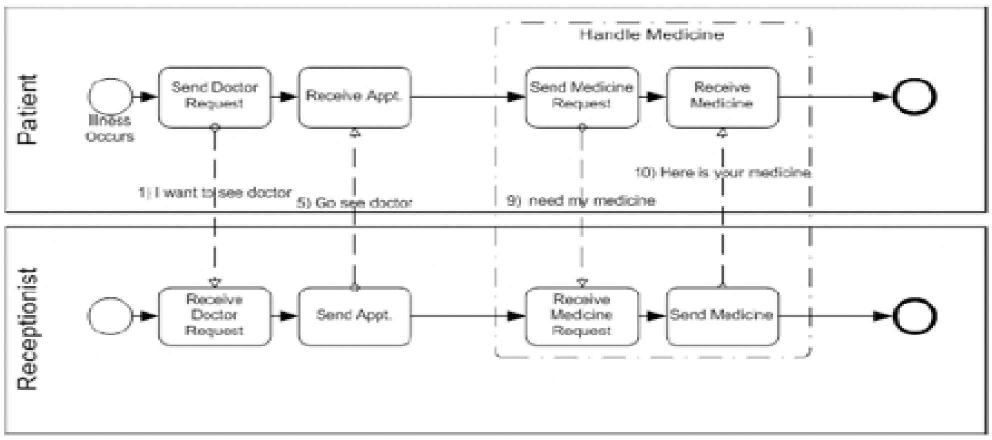
- it requires training to understand the notation of the graphical technique (Shneiderman 1998)
- using a technique will not fully describe the environment of the computerized system.

- using a technique will lead to a model that to some extent describes the functionality of the computerized system

As described by the OMG (Miller and Mukerji 2003), graphical modeling techniques that should be used in creating a CIM should be directed to organizational/business aspects and also be used by, in first hand, the domain practitioners/end-users but the process oriented modeling techniques has a notation with quite high formalism. Symbols are described (boxes, circles, diamonds) and different type of arrows (filled, dotted, double arrows etc) are also described. In order to use the process-oriented modeling techniques in CIM the domain practitioners need training and education about the notation (Shneiderman 1998).

BPMN has attracted considerable attention in the IS research field as a convenient description technique for documenting and re-engineering processes (Recker, Indulska et al. 2006). BPMN consists of the following basic building blocks (Figure 2, to the left) (White 2004).

- Flow objects: events (circles), activities (rectangles with rounded corners), and gateways (diamonds)
- Connecting objects: mainly comprising arrows, these indicate sequence flow (filled arrows), message flow (dashed arrows), and associations
- Swimlanes: pools (graphic container) and lanes (sub-partition of the pool)
- Artifacts: data objects, groups, and annotations



BPMN

Figure 2. Examples of BPMN (White 2004)

UML-activity diagrams are typically used for business process modeling, for modeling the logic or for modeling the detailed logic of a business rule (Figure 3). In many ways UML activity diagrams are the object-oriented equivalent of flow charts (Ambler 2004; Ambler 2007).

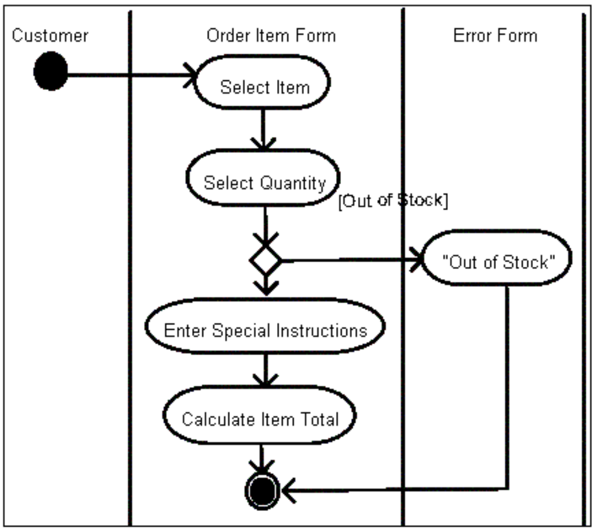


Figure 3: Example of UML-activity diagram

The main reason for the popularity of **EPC** is that it is a component in the SAP Reference Model (Aalst 1999.; Mendling, Aalst et al. 2006). The SAP Reference Model is a set of information models that is utilized to guide the configuration and implementation of SAP-systems.

The SAP Reference Model contains almost 10000 individual models with 3000 of them being EPCs (Mendling, Aalst et al. 2006). A major part of these models is business process models represented in the Event driven Process Chains (EPC) notation. EPC comprises the following elements (Figure 4) (Aalst 1999.):

- Functions: these building blocks correspond to an activity (task, process, or step) that needs to be executed.
- Events: these describe the situation before and/or after a function has been executed; functions are linked by events.
- Logical connectors: these can be used to connect functions and events to specify the control flow; there are three types of connectors: Λ (and), XOR (exclusive or), and ν (or).

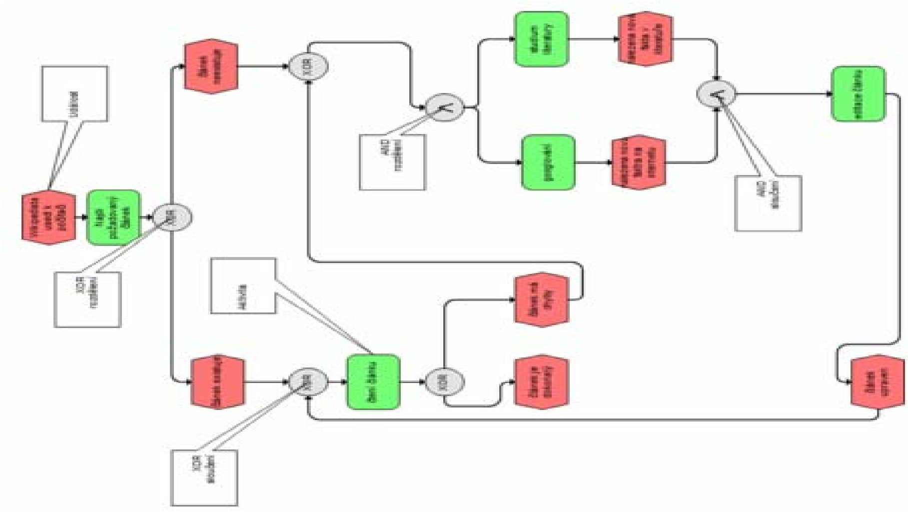


Figure 4: Example of EPC

4 Analysis, results, and discussion

The analysis is based on the important findings in the quality criterions presented by Moody (2006), experiences from research projects and educational settings related to creating computer independent models. We evaluated each of the four identified modeling techniques according to Moody quality criterions. We used a three graded scale: ☺, 😐 and ☹. ☺ = +1, 😐 = 0 and ☹ = -1.

The analysis started with a discussion about how to interpret and use the scale. We focused on the contrast between the different values in the scale. In order to give the reader a more thorough understanding of the analysis (and the use of the scale) we will present two examples from our discussion and analysis.

The first example relates to the researchers' understanding of the concept *discriminability* in the process-oriented modeling techniques. If we have a model with boxes and connecting arrows/lines (frequently used in ISD) we looked on the arrows to see if there was a description on the arrow and a direction (☺ = +1), if the arrow did not have the description but had a direction (😐 = 0), or if the arrow did not have both a description and a direction (☹ = -1).

The second example relates to the researchers' understanding of the concept *perceptual and cognitive limits* in the process-oriented modeling techniques. This started with the question: How well does the modeling technique support decomposition and modularization? If the modeling technique supports decomposition and modularization (☺ = +1), if the modeling technique partially supports decomposition and modularization e.g. splitting and merging (😐 = 0), or if the model does not support decomposition and modularization (☹ = -1).

This is two examples of the interpretation of the concept presented by Moody. Our analysis followed the logic in the described examples.

Three researchers did the analysis and classification independently. The outcomes were compared. There was more than an 80% agreement between the three outcomes, which is an acceptable inter-rating coder reliability. Where non-agreement existed, the three researchers discussed the evaluations and final classification and evaluation decision was made.

The result from the analysis is presented in Table 1. The rows in the table are each one of the nine criterions presented by Moody and the columns are the four process-oriented modeling techniques.

	Flowcharts	BPMN	EPC	UML activity diagram
Discriminability	☹	😐	☹	☹
Perceptual and cognitive limits	😐	☺	😐	😐
Emphasis	☹	☹	☹	☹
Cognitive integration	☹	☺	😐	☺
Perceptual directness	😐	😐	😐	😐
Structure	☹	☺	☹	☹
Identification	😐	😐	😐	😐
Expressiveness	☹	☹	☹	☹
Simplicity	☺	☺	☺	☺
Sum:	-4	+2	-3	-2

Table 1: Results of the analysis

4.1 Discussion

The discussion will be structured according to our results presented in Table 1. The maximum sum that could be reached for the process-oriented modeling technique was 9 and the minimum was -9. Most notable is that only one of the modeling techniques (BPMN) has a positive sum (+2). The other three modeling techniques have a negative sum. Based on the used quality criterions, this indicates that there are problems to be solved when it comes to modeling techniques. BPMN reached the highest sum (+2) but it is far from the maximum sum (+9) which indicates that even BPMN could be improved regarding. All four process modeling techniques have a negative value (☹**)** on *emphasis and expressiveness*, a neutral value (😐) on *perceptual directness* and *identification*, and a positive value (☺**)** on *simplicity*. These findings indicate that a rethink about process modeling techniques as a tool for computer independent models is needed. First and foremost process-oriented modeling techniques must start using visual variables to reach a higher expressiveness and emphasis (as described in Section 2). The different process-oriented modeling techniques used more or less the same graphical symbols (variants of boxes and arrows) over and over again and there is no way of telling which are the most important objects/processes in a model.

During the literature review we were also surprised by the fact that:

- there were very little written about CIM and models. Most of the examples in the literature were descriptions about automatic or semi-automatic transformations/mappings between PSM and code (Wärja 2007)
- issues of semantics and content (what constructs to include in a notation) are treated as matters of substance, but details of graphical syntax (how to visually represent these constructs) are treated with no or little consequences
- many graphical notations in ISD are not consistent with principles of graphical design (visual appearance or form)
- graphical models are treated as stand alone artifacts.

5. Conclusion and future research

When we started the research process the underlying research question was: Which are the relevant graphical modeling techniques to be used when creating computer independent models in the MDA framework? The answer to that question is that we had four candidates of process-oriented modeling techniques: flowcharts, BPMN, EPC, and UML-activity diagrams.

They satisfy (to some extent) the seven demands/requirements put forward by OMG. Using Moody’s quality criterions to analyze the models, only BPMN reached an acceptable level. There are strong indications that computer independent models should benefit from not only to be discussed from issues of semantics and content (what constructs to include in a notation) but also details of graphical syntax (how to visually represent these constructs) and how well they serve as communication catalyst in the MDA-architecture. One major reason behind this statement is that computer independent models should first and foremost are used by domain practitioners as a communicative tool and for these models the principles of good graphical design should be applied.

The findings presented in this paper will be evaluated in another empirical driven research project. A research project has been initiated together with two software development companies in the southern part of Sweden that has adopted the MDA. The research project will be implemented with the software development companies together with their clients. The research partner has accepted the use of a mix of graphical modeling techniques in order to produce a CIM based on Moody's graphical principles. We will start to use symbols/icons (from pictograms), colors (red, yellow and green from Lean Design), sizes (relatively each other) together with a graphical process-oriented modeling technique in order to produce a number of working hypothesizes that will be evaluated later on in the project and in the end maybe a more expressive modeling technique will be developed.

6. References

Aalst, W. M. P. v. d. (1999.). "Formalization and Verification of Event-driven Process Chains." Information and Software Technology **41**(10): 639-650.

Ambler, S. W. (2004). The Object Primer 3rd Edition: Agile Model Driven Development with UML 2, Cambridge University Press.

Ambler, S. W. (2007). "Introduction to the Diagrams of UML 2.0." Agile Modeling Retrieved 2008-03-10, 2008, from http://www.agilemodeling.com/essays/umlDiagrams.htm.

Johansson, L.-O., B. Cronquist, et al. (2007). KNOWLEDGE AND COMMUNICATIVE ASPECTS OF VISUALIZATION IN ACTION CASE RESEARCH. User Interface and Human Computer Interaction 2007 (Part of MCCSIS 2007), Lisboa, Porugal.

Johansson, L.-O., B. Cronquist, et al. (2007). Visualization as a tool in action case research. 6th European Conference on Research Methods in Business and Managment, Lisboa, Portugal.

Mathiassen, Munk-Madsen, et al. (2000). Object oriented analysis and design, Studentlitteratur.

Mendling, J., W. v. d. Aalst, et al. (2006). "Errors in the SAP Reference Model." BP Trends.

Miller, J. and J. Mukerji. (2003). "MDA Guide." Retrieved 2008-02-29, 2008.

Moody, D. (2006). What makes a good diagram? Improving the cognitive effectiviness of diagrams in IS development. ISD 2006, 15th international conference of Information Systems Development, Budapest, Hungary, Springer.

Recker, J. (2006). Towards an understanding of process model quality. Methodological considerations. ECIS 2006, Gothenburg.

Recker, J., M. Indulska, et al. (2006). HOW GOOD IS BPMN REALLY? INSIGHTS FROM THEORY AND PRACTICE. ECIS 2006, Gothenburg.

Selic, B. (2003). "The pragmatics of Model-Driven Development." IEEE Software(September 2003): 19-25.

Shneiderman, B. (1998). Designing the user interface, Addison Wesley Longman.

White, S. A. (2004). "Introduction to BPMN." 2006, from http://www.bpmn.org/Documents/Introduction%20to%20BPMN.pdf.

Vidgen, R., D. Avison, et al. (2002a). Developing Web Information Systems, Butterworth-Heinemann.

Wärja, M. (2007). Model Driven development: a review of current research. 6th International conference on perspectives on Business Informatics Research, Tamperee, Finland.