Minimizing the time spent on transformations between computer independent models and platform independent models

Harald Kjellin, Lars-Olof Johansson, Magnus Wärja
Stockholm University, Kristianstad University, Lund University
harald.kjellin@hkr.se, lars-olof.johansson@hkr.se, magnus_warja@hermes.ics.lu.se

Abstract. A well known trade-off effect with modelling is that when it is user friendly it becomes ambiguous and when it has a formal notation, that makes it unambiguous, it becomes less user friendly. A well known solution to this dilemma is to use models on various levels of formalism where the least formalized models are known to be more user friendly and the most formalized models are well suited for being directly translated to program code. This article investigates how the user friendliness at an informal level can be combined with the accuracy needed at a more formal level of modeling. It aims at combining an informal user friendly perspective with a formal developer perspective in order to **minimize the time spent on transformation mapping** between user friendly models and developer models. Theoretical data and empirical data indicates possibilities of minimizing the work with transformations between models. The results show that a number of developed criteria can be used to secure the user friendliness of models that are used in communications between the designer and the end user.

1. Introduction to levels of modeling

Model Driven Architecture (MDA) as it is presented by the Object Management Group (Miller and Mukerji 2003) describes how graphical models are used on three different levels (CIM, PIM and PSM see below) where each level is using models at a certain abstraction level.

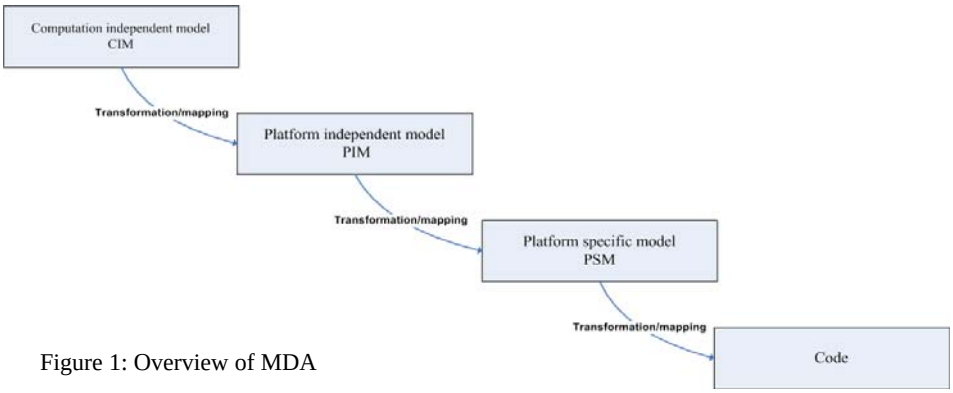

Figure 1: Overview of MDA

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 (see Figure 1 at the previous page).

In another article about Graphical modeling (Johansson, Wärja et al. 2008) there is a description about the usability of various graphical modeling techniques for MDA. The focus is on to what extent they satisfy the quality criteria described in (Moody 2006). These criteria have a focus on the user friendliness of the initial stages of modeling often referred to as Computer Independent Modeling (CIM).

The results in the article (Johansson, Wärja et al. 2008) indicate that the modeling approaches “Rich Pictures”, (Checkland 1981; Checkland and Poulter 2006) and “Business Process Modeling Notation” (BPMN) (White 2004; OMG (ObjectManagmentGroup) 2008; Recker, Indulska et al. 2006) are best suited for communication with end users in the initial phases of MDA.

2. Purpose

The purpose with this paper is to find which type of modeling is best suited for combining the user friendliness of informal models like “rich pictures” with more syntactically uniform and formal modeling languages in a way that facilitates or **minimizes the time spent on transformation mapping** between the various levels in both directions, i.e. both from CIM to PIM and also from PIM to CIM.

3 Problems with transformation mapping between models

There seems to be a number of various criteria for what constitutes a good graphical description in general and the article by (Johansson, Wärja et al. 2008) clearly describes the best approaches to constructing a model on the highest level in MDA, the CIM–level.

We have, however, found that neither the article (Miller and Mukerji 2003) or related articles contain enough useful instructions for how the transformation mapping between levels of modeling shall be carried out. There exist a number of publications concerning how to transform the PSM to code, but no articles have been found that does any analysis of all other types of transformations and especially not the backward transformations to previous less formalized levels. When trying to do such mapping manually there is often much work needed.

Today much design of computer systems is done in an iterative way. For example “fast prototyping”, “rapid prototyping”, “agile development”, “extreme

programming", etc. are all well known approaches that require that the developer continuously communicates with the end user/client/customer/domain expert (Beck 2000; Jeffries 2001; Ambler 2004). In this continuous communication with end users and other developers it would be necessary to make sure that all new additions to the code would also be reflected in the models on higher levels. If this would not be the case the models would quickly become obsolete and would not at all correspond with the real system.

History about documentation of program code shows that this type of neglect is more usual than it is not, which is probably the main reason why modeling is not used more than it is. The question is then: Is it possible to apply modeling in such a way that as few transformation mappings as possible are needed between the various levels of models and especially between the models CIM and PIM.

A similar relevant question in this context is whether transformations really are needed. One could argue for the opposite, i.e. that there may be less work for the developer if various models could remain independent of each other? However, if this was done the developer would have to improvise descriptions of the system with the end user without being certain that all functions in the system had been talked about in the improvised dialogue with the end user. The reason the developer would have to improvise is that the end user would not benefit from being shown the formal technical notations used by the developer and the developer would have to redesign the user friendly CIM model without being able to check if all parts of it had been modified according to the latest version of the PIM model. To put it simply: Without an efficient transformation mapping between CIM and PIM and vice versa between PIM and CIM there is a high risk that there is not enough high quality communication between the developer and the end user.

3.1 The problem with transformation mapping in two directions

If the graphical models are to be user friendly for end users that are not computer or modeling experts it is difficult to make them useful when exact modeling is needed in system design. Assume, for instance, that the end users have accepted a description in the form of a "rich picture" and then the system designer converts this to a more formal model before s/he wants to implement it. While working with this, the developer wants to communicate some changes or alternative solutions that may be needed. In this case the developer may have to reconvert the formal models into less formal models in order to make sure that the user can understand these models. This means that for every meeting with an end user the developer must a) convert the formal model into a less formal one, b) discuss the changes with the end user, and c) convert the discussed changes back into a formal model to make sure that all people working with them understand them in an ambiguous way. In real developments few developers would feel comfortable with the extensive work associated with continuous transformations between

various types of models. There is thus a risk that the modeling work would result in errors in the models.

3.2 The risk with enforcing end users to understand a PIM model

Another approach to the problem is to try to educate the end users in how to use the formal language of the PIM model. This would make the CIM model obsolete and would imply that the developer could speak more freely with the end user. This approach may work well with some end users but there would always be a risk that the end users do not understand the models and do not want to communicate about their possible misunderstanding, which would also lead to possible errors in the models.

3.3 The problem with mixing perspectives

Graphs that mix two or more perspectives are much more difficult to grasp than graphs that display a single perspective. An example of a mix of perspectives is when a graph mixes a static perspective like in a taxonomy or hierarchy with a temporal perspective like in process descriptions. We have found (see experiments described later on in this article) that when perspectives are mixed the user needs much more time to distinguish between the various perspectives in the graph. This also makes the transformation mapping between one graph to another graph much more difficult.

3.4 A problem with Daniel Moody's criteria on model quality

The criteria for user friendly models described in (Moody 2006) are on a high level of abstraction. This is the work that is most similar to the work described in this article. The main difference between Moody's work and our proposed criteria is that we aim at defining more specific criteria that minimizes the time needed for transformation mapping between models. Moody secures the usability and user friendliness of the models, but his criteria does not secure an efficient transformation mapping between models.

4. Method for developing criteria that facilitates transformation mapping

We classify our research approach as an inductive generation of hypotheses. The work is focused on criteria that are useful for supporting transformation mapping between informal and formal models. It should be seen as a pilot study for generating criteria to be further tested and developed in experiments with

modeling. Thus we do not aim at verifying the usability of the criteria but we are restricting our method to only verify their relevance.

Initially we combined existing criteria extracted from related research with general theories of user friendliness in order to develop an initial body of criteria. Then we refined these criteria incrementally by testing and verifying them against an existing body of classified examples of models

4.1 Requirements on the initial criteria for evaluating models

We based the work on two sources. The general quality criteria for graphical models described in (Moody 2006) and the analysis of how various graphical modeling methods satisfied these criteria as was described in (Johansson, Wärja et al. 2008). This gave us indications about the drawbacks in graphical models. We did realize that this was not enough to enable us to design some solution to the problems previously described. We therefore developed a set of initial criteria by combining criteria and theories from various sources as described below. We focused on criteria that can be used to minimize the work with transformation mapping.

We started out from some of the results concerning “Rich pictures” and “BPMN” described in (Johansson, Wärja et al. 2008). These criteria to be developed should satisfy the following requirements:

1. They should conform to the general criteria described in (Moody 2006) and also described further on in this article
2. They must be easily applied
3. They should result in graphical descriptions that are easily understood by those of the end users that have little experience of modeling
4. They should be applicable on abstract informal levels as well as on more specific computer dependent levels, i.e. there should be no need to perform any substantial transformations between levels of abstraction or degrees of formalization.
5. The only transformations necessary should be restricted to adding notation at more specific levels or taking away specific notation when going from specific models to higher level models. Any fancy symbol or illustration at the CIM – level could remain at the PIM level as long as it did not prevent the interpretation of the formal syntax at the PIM level.

4.2 Method for specifying who should use the criteria

The developer in charge of system analysis or system design should also be the person that is in charge of the transformation mapping between the formal models

at the PIM level which may contain a difficult syntax and the informal models at the CIM level who are there to facilitate communication with the end user. This means that the criteria have the developer as its target user. It is the developer who needs to convince the end user that what s/he sees in the informal graph is also what s/he will get in the completed system. It is the developer who continuously needs to transform difficult formal models to simple informal ones in order to enable an easy communication with the end users, and then when the communication has resulted in a modification of the informal model the developer needs to convert it back to a formal model again to secure that its integration into the system is syntactically correct. Once any modification is transformed from the CIM to the PIM it probably needs to be correct enough to be able to function as a specification for program code. A metaphor for describing how the criteria is intended to be applied is that they can be used to add restrictions to “Rich Pictures” in order to decrease the necessary work when transformation the “Rich Pictures” back to PIM models. To impose such restrictions will initially create some more work for the developer but we assume that it will create less work in the long run since the transformation mapping will have to be done each time the developer checks new any newly implemented function with the end users.

4.3 Selecting method for refinement of the initial criteria

The difficulty with finding new criteria, without having any existing verified criteria to start out from, convinced us that an incremental approach to developing them would be the most rewarding method. The incremental approach consisted of a continuous iteration of defining, evaluating and refining the criteria until we had enough indications that they could meet the requirements described in 4.1 above. We assumed that if we test the existing criteria on already existing models that were classified as being either on the CIM level or the PIM level, then this would produce test data that could give further indications about the criteria.

5. Finding drawbacks and advantages in existing graphical descriptions

In order to find enough models that were somewhere between the CIM level and the PIM level we decided to focus on models created by non-experts. We needed to investigate which types of errors and shortcomings novice designers of models usually do in order to be able to create criteria that support designers in not doing these errors and at the same time fulfilling the general quality criteria for user friendliness of CIM described in (Moody 2006).

In order to find hints at how non-experts would model complex processes we selected 52 master theses, from a large set of stored master thesis, whose aim was not related to modeling but who were utilizing improvised models to describe complex phenomena. The intuition behind this was to find indications of advantages and drawbacks with models that were spontaneously created by students. We studied the theses in the following way:

1. We browsed through each of thesis and extracted the models we could find although we did not select any platform dependent models since these were outside the scope of the research.
2. The quality of the models described in each thesis were subjectively interpreted and classified. The purpose with this was to find often occurring weaknesses and strengths in models. Some models were quite informal (similar to rich pictures) and some were formalized to an extent that made them difficult to understand.
3. The most crucial weaknesses in the models were tested with five test subjects. The test gave confirmations concerning which parts of the models were difficult to understand and which were easy to understand.
4. We continuously re-defined the criteria to support the novice designer in not making the same error again. For each defined criteria we tested them against the requirement specified previously in section 4.1
5. We extracted and generalized the criteria that we estimated as being most useful.

The above method was carried out in an interpretative way. There was no intention to verify the usefulness of the criteria in a quantitative way. The only purpose was to make a preliminary sorting of possible quality criteria in order to constrain the complexity of further investigations. The final candidate criteria were then typed and sorted. They are described in the next section.

6. The developed criteria

6.1 General criteria to secure the communication between a designer and a end user

The general criteria is taken from (Moody 2006) and these can be used to secure the basic or general communication between a designer and a customer. We assumed that the approach called “Rich Pictures” is a good metaphor for describing the type of modeling that has few or no formal restrictions. Another metaphor for similar graphical modeling is called “Mind mapping”. The quality

criteria that could be associated with these types of models coincide with the criteria described in (Moody 2006):

1. Discriminability. To what extent does one diagram element differ from another diagram?
2. Perceptual and cognitive limits. Does the diagram contain so many elements and relationships that it is hard to get an overview?
3. Emphasis. Can you see what is important and what is not important?
4. Cognitive integration. Is it easy to relate various types of diagrams with other types of diagrams?
5. Perceptual directness. To what extent does the model use pictures or symbols. To what extent are all parts of the picture conform to common sense. To what extent are basic spatial principles of human perception followed. Is there for instance a steady flow from one end of the graph to the other?
6. Structure. If there is much information in the diagram is it divided into groups to facilitate understanding without exceeding cognitive limits?
7. Identification. Is there a name on every link? Is there a name on every picture/drawing/diagram element?
8. Expressiveness. Is a full range of visual variables used?
9. Simplicity. To what extent is it easy to redraw the model and thus allow for continuous redesign when discussing the model?

6.2 We have added some criteria concerning the typing of the symbols in the “rich pictures”

The problem with rich pictures is that it may become difficult to communicate about what each part of the picture specifically refers to. To decrease this difficulty it may be necessary to introduce a soft type of a general formalism in order to decrease the complexity of unrestricted rich pictures. The ambition is to decrease the risk of misunderstandings concerning the meaning of the parts and links in the models. The quality criteria were:

- To what extent are there distinctly specific types of links between nodes in the picture? For instance specific notation on the links describing:
    - The sending information between nodes
    - Static relationships between nodes
    - Causal/commands/control relationships
    - Steps, where the linking of nodes shows in which order they occur
- To what extent are there distinctly specific types of headers on the pictures, for instance:
    - Living agent, Buyer, Seller, End user, etc…
    - Process or activity
    - Static structure, data, information

- If a relationship between two pictures is two-ways it may be obvious what is represented in each of these, but sometimes the meaning of the relationship is only obvious for its creator. (We found many examples of general links that could be easily misunderstood). The requirement says that it should preferably be described as two separate relationships with a specific name for each, even if the name is the same for both.
- To what extent is the grammar/legend directly associated with the first version of the graph. This secures that nobody is lost because s/he does not know what any notation means.

6.3 Comment concerning conversions between different levels of formalization

We have assumed that the only reasonable way to apply the presented criteria is to never initiate any conversion of the graphic description when going from a lower level to a higher level. Instead the developer should minimize the needed conversion by sticking to a single perspective even when s/he moves from a lower level of abstraction to a higher level of abstraction. The simplicity and lack of formal symbols should in itself guarantee that the end user will understand what is modeled.

6.4 We have added some criteria about avoiding the mixing of perspectives

The following criteria are aimed at making the “Rich picture” as simple as possible. They are also aimed at securing that the “Rich Picture” does not contradict more formal computer dependent models at the level PIM.

- To what extent does the model avoid the mixing of a perspective with another perspective? (The models that followed only one single perspective were usually the best). For instance perspectives like:
    - Taxonomies (Object oriented modeling)
    - Structures that divides the whole into parts where all links describes the subparts of any part of the structure
    - Command structures (sometimes called report structures) that describes a hierarchy of authority
    - Memory schemas. Like conceptual models for designing a database
- It is very important that each perspective contains links to other parts in other models that are designed from another perspective. This is needed to facilitate for the end user to understand the relationship between the various models

6.5 We have added some criteria about process descriptions

Process models are probably the most common types of models throughout the history of modeling in computer science. In order for these to be as complete as possible we propose some more criteria:

- Is it clear which super process is the process a part of
- Is it always clear where the beginning is and where the end of processes and sub processes are.
- The input to each process should be of a standard type depicted as a name
- The output should be of a standard type depicted as a name
- Explicit triggering of a process. Does the model describe which combination of input is necessary to trigger an activity that in turn sends an output to another part of the model?

The criteria above may add complexity to the graph that violates Moody's (2006) criteria about "Cognitive Integration". We do, however, assume that the criteria can be implemented in alternative ways to avoid this. For instance by adding footnotes to textual descriptions located directly below the graph.

7. Results from experiments with the generated quality criteria

After analyzing and testing the criteria for minimizing the time needed for transformation mapping we have assumed the following:

- The main goal with using CIM modeling is to provide a user friendly overview of the work which remains consistent with work on more specific levels
- The proposed criteria will impose general restrictions that will secure that the user friendliness of the model can be combined with more formal models. Especially since the use of a singular perspective in a graph will conform to classical ways of modeling outside the world of computer science
- The criteria will secure that fewer transformations will be needed between CIM and PIM models
- The enforced typing decreases the risk of misunderstandings and secures that less work will be needed on transformations
- The enforcement on adding descriptions on input and output from processes secures that the model is more easily transformed between CIM and PIM
- The enforcement on descriptions on conditions on what triggers processes also secures that there are less transformations between the levels

Unfortunately there is not room enough in this article to provide examples of how the criteria can be applied on various types of graphs, which implies that there is no clear verification of the usefulness of the criteria. This should be done in future research for verifying the hypotheses described in this article.

8. Discussion

The purpose of the presented research was only to arrive at a number of criteria to be further tested and verified. Apart form these criteria we have some informal and speculative ideas that we intend to develop further by comparing the resources needed for modeling in relation to the benefits of the modeling at each of the described level. We assume that a Return of Investments (ROI) study would show that:

- The major strength of visualizations is to give an initial overview of a situation. This implies that visualizations are preferable in the initial part of the work that is often carried out on a level where it is necessary to depict the large number of relationships between general abstractions
- We found indications that the concept of ontology in computer science is a much too difficult concept to be applied in dialogues with end users. We assume that it is more rewarding to only discuss one single perspective at a time. If ontology is used it should only be used for the internal dialogue between developers

When we studied which of the models from the article (Johansson, Wärja et al. 2008) matched the criteria we discovered that the criteria could possible be added to the “Rich Pictures” or the “BPMN” notation.

9. Conclusion

We have described a set of hypotheses concerning quality criteria for model driven architectures that will enable the developer to make more efficient transformations between the CIM and PIM level. **The results from informal experiments with implementing the criteria indicate that the criteria minimize the time needed for transformation mapping**. We intend to proceed with implementing the criteria in an extensive number of example models and then compare these with well known models. The comparisons will be done in experiments with human subjects. The aim with this will be to verify the usefulness of the criteria.

References

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

Beck, K. (2000). extreme programming explained, Addisson Wesley.

Checkland, P. (1981). Systems Thinking, Systems Practice, John Wiley & Sons.

Checkland, P. and J. Poulter (2006). Learning for action:a short definitive account of Soft Systems Methodology and its use for Practioners, Teachers and Students. Hoboken, Wiley.

Jeffries, R. (2001). "What is Extreme Programming?" Retrieved 2004-04-13, 2004, from http://www.xprogramming.com/xpmag/whatisxp.htm.

Johansson, L.-O., M. Wärja, et al. (2008). Graphical modeling techniques and usefulness in the Model Driven Arcitechture: Which are the criteria for a “good” Computer independent model? Under Consideration.

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.

OMG(ObjectManagmentGroup). (2006). "Business Process Modeling Notation Specification." Retrieved 2008-02-29, 2008.

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

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