Modelling: From Craftsmanship to Automation

Harald Kühn[1], Marion Murzek[2, *]

[1] BOC Information Systems GmbH, Rabensteig 2, A-1010 Vienna, Austria
harald.kuehn@boc-eu.com

[2] Women's Postgraduate College for Internet Technologies (WIT),
Vienna University of Technology, Favoritenstr. 9-11/E188, A-1040 Vienna, Austria
murzek@wit.tuwien.ac.at

Abstract. This paper investigates the evolution of the automation process of modelling in the context of enterprise information systems. An evolution schema for modelling automation based on the dimensions “process”, “artefacts” and “technology” is introduced. Modelling currently experiences a similar evolution of automation such as other disciplines, like architecture, automotive, and trade have done in the past. We examine “why” and “how” things change in modelling automation, and we try to provide a forecast for future developments. Three distinct automation levels of modelling processes – manual, guided and automatic – are presented and underpinned with real life examples.

1 Introduction

Modelling is used to describe the relevant aspects of an original under examination. Models therefore facilitate the communication between people by concentrating on the essentials of a problem under consideration.

Conceptual modelling is accepted as one of the core disciplines within business informatics [11]. The scope of the presented work lies on the evolution of modelling automation in the area of enterprise information systems. The models in this area can be roughly separated in models supporting (1) the software development process, (2) the software development itself and (3) a company’s business. *Software development process* models are procedure models to guide the developer through the development process, e.g. waterfall model, spiral model or unified modelling process. With *software models* the design of software can be graphically illustrated (e.g. UML), and partly or sometimes be simulated. Often it is even possible to partially generate code out of software models. One initiative in this area is the model driven architecture (MDA) proposed by the OMG [33]. *Business models* help to represent a company’s structure and behaviour. The organisation structure represents the people, depart-

* This research has been partly funded by the Austrian Federal Ministry for Education, Science, and Culture, and the European Social Fund (ESF) under grant 31.963/46-VII/9/2002.

ments, roles, infrastructure, and interrelations between them. Business process models described the behavioural aspects of an organisation.

As in the areas of architecture, automotive industry, trading etc. there was and still is evolution in automation in the modelling domain. The evolution of modelling automation starts with manually drawing models with a pencil on a sheet of paper and may culminate in the automatic generation of models.

Compared to other disciplines such as trading, the evolution of modelling automation in the area of enterprise information systems has occurred in a very short period of time. The driving forces for this rapid development were, among others, the number of heterogeneous implementation platforms, the growing complexity of enterprise systems, and the rapidly increasing interdependencies between enterprise systems.

So how have these problems influenced and been addressed by the development of modelling? With the evolution of modelling, the power of expressiveness has increased. Standards support a common understanding of facts and help to accelerate corporate work. The evolution of modelling produced techniques that support the user in creating models, e.g. procedure or reference models. As software projects often suffer from lack of early evaluations before operation, mechanisms such as simulation of software models help to reduce the associated risks, and optimizations can be performed earlier. By means of business process modelling, processes could be optimized, costs could be reduced and individual tasks could be simplified and partly automated.

In chapter 2 an evolution schema of modelling automation is introduced. Chapter 3 examines modelling evolution and identifies three general evolution steps. Chapter 4 describes three modelling automation processes – the manual, guided and automatic processes – and underpins them with real life examples. Related work is provided in chapter 5, and chapter 6 gives a summary and outlook to future work.

2 Evolution of modelling automation

The evolution of modelling in the area of enterprise information systems began approximately 70 years ago and can be subdivided in the evolution of the *modelling process ("how")*, the *modelling artefacts ("what")* and *modelling technology ("with what")*.

Criteria/ Time	< 1940	~1940	~1950	~1960	~1970	~1990	> 2000
Modelling Technology	Pencil & Paper	Flow-charting templates	Textual Modelling Software	Modelling Software	CASE Tools	Metamodelling Platforms	Query Supported Tools
Modelling Artefacts	Drawing	Lines & Shapes		Predefined Elements		Predefined Patterns	Generated Models
Modelling Process	Manual			Guided		Automatical	

Table 1: Evolution Schema of Modelling Automation

The interrelation of these three influencing factors made modelling what it is today, a semi-automated and in some areas a fully automated discipline, e.g. automatically generated infrastructure models.

2.1 Process

A process is the way of how to do something. On the one hand, the process is supported by artefacts and technology. But on the other hand, the process influences the evolution of artefacts and technology.

Until the ‘60 models were exclusively drawn by hand. With the invention of the computer, it became possible to draw models electronically. Without guidelines and restrictions, it was possible to create models on the computer. In the middle of the ’60, as the *procedure model* arose, the modeller was guided through the modelling process. Well-known examples for such procedure models in the area of systems development life cycle models are the waterfall model [36], the spiral model [6] and the unified modelling process [22]. Between the ’60 and the ’90 the next level of user support was invented - the *reference model*. A reference model is a “sample model” which could be adapted by the modeller. A well-known example for a reference model is the Information Technology Infrastructure Library (ITIL) for the IT Service Management domain [19]. The adaptation of such reference models is supported by specialized procedure models. Nowadays, the modelling process has begun to work *automatically*. An initiative for this advancement is the model driven architecture (MDA) [33]. In the area of infrastructure modelling for example, models could be generated out of the automatically retrieved information of a company’s system infrastructure.

2.2 Artefacts

Modelling artefacts are used to make models visible. They change with the need for more power of expression. The first typical modelling artefacts after using simple *lines and shapes* were *predefined modelling elements*. One of the first standardized diagram type was the dataflow diagram (DFD). Beside other elements it uses rectangles for expressing a process and ellipses for drawing start and end elements (fig. 1) [39].

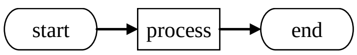

Fig. 1: Example of Data Flow Diagram

Predefined modelling elements evolved to entire modelling languages. They provide ready-made elements for models in different kinds of domains. Sometimes the models of one domain are separated in different diagram types with the possibility to link them together. Around 1970, two kinds of modelling could be distinguished: modelling of structure and of behaviour. One of the first “standards” for structure modelling were the Bachmann diagrams [5], invented in 1969. Another modelling language, the Entity Relationship diagram [12], arose with the paradigm of relational databases approximately in 1972. The first modelling language for behavioural modelling were the Petri Nets [35], invented by Carl Petri in 1963. Ten years later in 1973 Nassi Shneiderman published the flow diagrams [31]. Today the most well known standard for structural and behavioural modelling concerning software development is the Unified Modelling Language (UML) [32], which was brought to being by a request for proposal (RFP) by the OMG in 1995. Their ancestors were Object Analysis and Design (OAD) [7,8], Objectory [21], and the Object Modeling Technique (OMT) [37].

First approaches concerning business modelling standards can be observed in 1977 in the work of Zismann "Specification and Automation of Office" [40]. Another approach was invented by Ellis in 1978: the Information Control Nets (ICN) [16]. In 1982, Dennis Tsichritzis invented the Form Flow Systems [38]. He distinguished for the first time explicitly between control flow and data flow. Further business modelling languages are Role-Activity-Diagrams (RAD) [34], Event driven Process Chains (EPCs) [25] and the Business Process Modelling Language (BPML) [10].

A further step in the history of modelling automation was the introduction of *predefined modelling patterns*. These patterns are collections of connected model elements which can be combined to create models. The latest achievement in the area of modelling artefacts is that of automatically *generated model elements*. Therefore, predefined patterns are used as well, but they can be generated and arranged automatically.

2.3 Technology

The modelling technology evolved with the requirements that appeared in the modelling process. For a long time, the easiest way to draw a model was with a *pencil on a sheet of paper*. A revolutionary step was the introduction of the first physical flowcharting template by IBM in the ´40 (fig. 2). This flowcharting template provided shapes and lines to draw diagrams on paper.

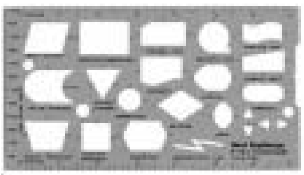

Fig. 2: Flowcharting Template

In the ´60, the first flowcharting program "Autoflow" was developed [23] by IBM. With this invention it became possible for the first time to draw dataflow diagrams on the computer. Simple *textual drawing software* provided the possibility to design models of any kind. The first *modelling tools* for enterprise information system models provided predefined modelling elements for drawing domain unspecific models. This kind of modelling tool was comparable with Microsoft Visio [29] today. *CASE tools* offered a whole modelling language for a specific domain. Representatives of CASE tools are Rational Rose [20], ModelMaker [30], Together [9] etc. The next step in the evolution of modelling technology were *metamodelling platforms*. Well known metamodelling platforms for business modelling are ADONIS [3], MetaEdit+ [27] and Metis [28]. Regarding software modelling, the most famous open source platform today is the Eclipse Modelling Framework [17]. Metamodelling platforms provide the possibility to configure several modelling methods, and additional mechanisms and algorithms are provided for analysing, simulating or evaluating a model. Nowadays, the modelling technology is moving towards *query supported tools* which can automatically generate models by means of queries on existing models or by generating models out of model repositories.

3 Evolution Steps in Modelling Automation

This chapter investigates in more detail the evolution of automation in the modelling discipline. Three "evolution triggers" can be identified (see table 2).

Trigger	Modelling Process	
	From	To
Standardisation	Manual	Manually supported by Lines & Shapes
Institutionalization & Specialisation	Manual supported by Lines & Shapes	Guided
Automation	Guided	Automatical

Table 2: Triggers of Evolution Steps in Modelling Automation

3.1 Standardization

The first development step in the modelling process (see table 2) was driven by the varying kinds of models and the misunderstandings due to different interpretations, as well as the poor power of expression of the "free hand" models of that time. The first step in solving this problem consisted in the invention of the flowcharting template (fig. 2). With this, a common syntax for modelling was provided. This made the expression of models more powerful and supported common understanding for the models.

In summary, it may be said that standardisation leads to common and better understanding of things.

3.2 Institutionalization and Specialisation

The next step in evolution of modelling began with a change of technology. Due to different modelling domains, there was a need for different modelling languages supported by modelling software, CASE tools and metamodelling platforms. Because of fairly complex model descriptions, there was a need for procedure and reference models to support the modeller by creating models. The first standardised modelling languages provided in CASE tools offered predefined elements. Together with the procedure model, the syntax and the semantic of modelling languages were defined. A further achievement was the possibility of simulating and optimizing the models created in a CASE tool or metamodelling platform.

The relation between these steps could be summarized with institutionalization and specialisation. Moreover, the people/users get more and more support, and the power of expression increases.

3.3 Automation

The third and last "big step" observed in the evolution of modelling and trading is the step to automation. In the case of the modelling process, the invention of query supported tools semi-automatically generating models mark the beginning of automated modelling. This reduces the errors occurred with manual or guided modelling. The models also became more precise and are generated faster than by humans.

All in all it can be said that automation reduces misuse and inaccuracies. Also it makes processes faster and – in general – more comfortable for the users.

4 Modelling Process Approaches: Examples

In chapter 2 general modelling process approaches were briefly outlined. In the following, three modelling process approaches will be described in more detail accompanied by real life examples. The metaphors used are *human actor,* representing the modeller, and *automated actor,* representing an information system supporting the modeller during his modelling tasks.

4.1 Manual Modelling

Manual modelling is the classical approach in enterprise modelling. The human actor carefully studies the domain under consideration. While doing so, she or he follows a certain modelling methodology, mainly based on her or his experience. By using a modelling language such as ER, UML, BPMN etc., she or he creates a model representation of the domain under consideration (fig. 3, step 1). To acquire the model-relevant information, techniques such as workshops, interviews, questionnaires and the like are used. In a second step (step 2), she or he interprets the model to see if it properly represents its original and if it is usable for analysing the problem domain.

The manual modelling approach places high demands on the modeller, because the only source for ensuring good model quality is the modeller’s experience.

Fig. 3, right side, shows an ADONIS model representation of an application process of a service company in BPMN [3]. This business process model was created by applying the manual modelling process approach.

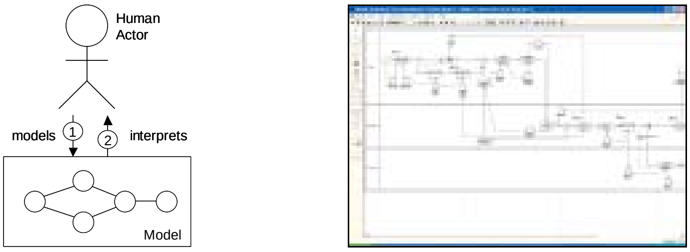

Fig. 3: Manual Modelling Approach

4.2 Guided Modelling

In the guided modelling approach, the human actor is supported by an automatic actor. Both actors participate actively in the modelling process. The automatic actor uses either a procedure model or reference models (fig. 4, step 1) to provide the human actor with support concerning the methodology and modelling language. Based on its configuration, the automatic actor guides and constrains the modelling tasks of the human actor (step 2). The human actor designs the models based on the input provided by the automatic actor and on her or his own experiences (step 3). The automatic actor checks if the model thus created conforms to the constraints of the procedure and the reference model (step 4). If not, steps 2-4 have to be repeated. Finally, the human actor interprets the model. The guided modelling approach makes less demand on a modeller’s experience than the manual modelling approach, because of the automatic actor’s support.

Fig. 4, right side, shows an ADOscore model representation of a Balanced Scorecard cause-effect-diagram. This cause-effect-diagram was created following a 7-step procedure model implementing the guided modelling process approach [4].

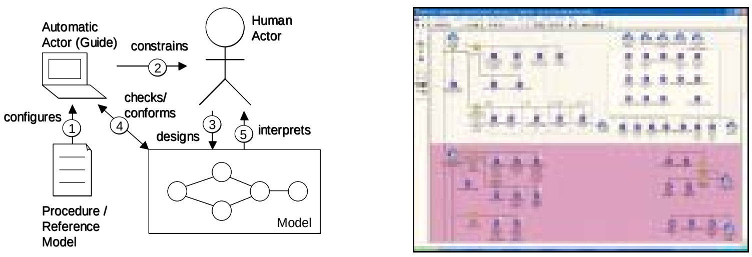

Fig. 4: Guided Modelling Approach

4.3 Automatic Modelling

In the automatic modelling approach, the automatic actor is the active part and the human actor is the passive part concerning model creation. The automatic actor is configured with rules concerning the queries of its model repository and other data sources, model creation rules and model layout rules (fig. 5, step 1). Based on these rules, the automatic actor generates new models automatically (step 2). Finally, the human actor interprets the generated models. The automatic modelling approach has currently found two major application areas: software engineering and IT architecture management. In the software engineering domain, model transformation based on MDA-like approaches targets automatically generated models for software development and system implementation [15]. In the IT architecture management domain, models such as application landscapes, operation models, network models and dependency models are generated automatically from data sources such as configuration management databases (CMDB), asset management databases and model repositories.

Fig. 5, right side, shows an ADOit model representation of a dependency model of a system infrastructure ranging from physical server nodes up to the supported business processes. This dependency model was created automatically by an automatic actor, which followed its configuration rules on querying the ADOit repository, generating a layered model representation and arranging the model elements. This example shows the automatic modelling process approach [2].

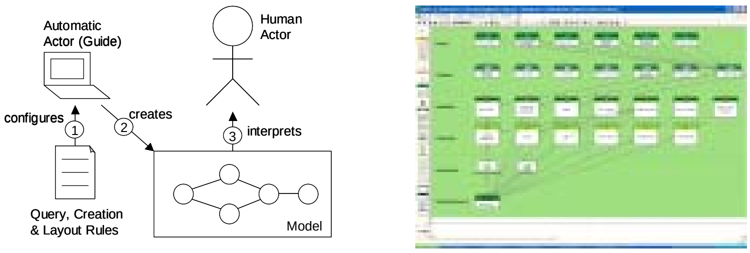

Fig. 5: Automatic Modelling Approach

5 Related Work

In a panel discussion at ER'97, historical perspectives and future directions of conceptual modelling were discussed [13]. A collection of papers concerning state-of-the-art and future direction in conceptual modelling can be found in [14].

An historical overview of object-oriented methods can be found in [24].

Model generation using model transformation is a key concept of MDA [33]. In [15], an extensive overview of model transformation approaches and categorizations into model-to-code approaches and model-to-model approaches is provided.

In the workflow management community, several approaches for the automatic reconstruction of workflow models from audit and execution data can found ("process mining"), such as [1, 18].

6 Summary and Outlook

Modelling currently passes through a similar evolution of automation as other disciplines such as architecture, automotive, and trade have done. We introduced an evolution schema for modelling automation and identified three general evolution triggers in modelling automation. By using real life examples from our project experiences in the enterprise modelling domain, we described three modelling automation processes: manual, guided and automatic modelling processes.

In the context of modelling automation we expect further research demand in the following areas:

- The acquisition and automatic transformation from tacit knowledge to explicit knowledge represented by knowledge models. First promising results in the acquisition of business process knowledge can be found in process mining approaches such as [18].
- In the MDA approach, models are generated along the axis “business description” down to “code representation”. Here, research in the domain of model transformations, model weaving and model synchronisation mechanisms preserving model semantics is necessary.
- To automatically generate models from different domains into a common representation, integrated domain metamodels are necessary. The Enterprise Model Integration approach (EMI) provides first results in this area [26]. Here, further research in metamodel integration and consolidation is necessary.
- In the IT Architecture Management domain, modelling automation is already applied in real life projects such as generating application landscapes, logical infrastructure and physical infrastructure models. Here, we expect further research applying agent technology for automatic model information acquisition.

References

1. Van der Aalst, W.M.P.: Process Design by Discovery: Harvesting Workflow Knowledge from Ad-hoc Executions. Online Proceedings of the Beyond Workflow Management: Supporting Dynamic Organizational Processes Workshop at the Conference on Computer Supported Cooperative Work 2000, December 2000. www.stern.nyu.edu/~abernste/cscw2000/papers/Process_Design_by_Discovery.pdf, access 2005-05-08.
2. ADOit Whitepaper. BOC GmbH, 2004.
3. ADONIS and BPMN Support. http://www.boc-eu.com, access 2005-05-08.
4. ADOscore Whitepaper. BOC GmbH, 2004.
5. Bachman, C. W.: Data Structure Diagrams, Data Base 1,2, Summer 1969
6. Boehm, B. W.: A Spiral Model of Software Development and Enhancement, IEEE, 1998
7. Booch, G.: Object-Oriented Development. IEEE Transactions on Software Engineering, SE-12 (2), 211-221, Februar 1986.
8. Booch, G.: Object-Oriented Analysis and Design with Applications (2. Ausgabe). Benjamin Cummings, 1994.

9. BORLAND Homepage /www.borland.com/together/, access 30 April 2005.
10. BPMI Homepage www.bpmi.org/BPML.htm, access 30 April 2005
11. Buhl, H.-U.; Heinrich, B.; Fettke, P.; Loos, P.; Frank, U.; Moody, D. L.; Parsons, J.; Rosemann, M.; Sinz, E. J.; Weber, R.: Empirical Research Strategies in Conceptual Modeling – Silver Bullet or Academic Toys? In: Wirtschaftsinformatik, Vol. 47, No. 2, 2005, Vieweg-Verlag, pp. 152-159.
12. Chen P.: The Entity-Relationship Model: Toward a Unified View of Data ACM Transactions on Database Systems I (I). March 1976, pp 8-46.
13. Chen, P. (Moderator): Panel Discussion “Conceptual Modeling: Historical Perspectives and Future Directions”. http://osm7.cs.byu.edu/ER97/er97prog.html, access 2005-05-08.
14. Chen, P.; Akoka, J.; Kangassalo, H.; Thalheim, B. (Eds.): Conceptual Modeling: Current Issues and Future Directions. Springer-Verlag, 1999.
15. Czarnecki, K.; Helsen, S.: Classification of Model Transformation Approaches. OOPSLA’03, Workshop on Generative Techniques in the Context of Model-Driven Architecture.
16. Ellis, C.A.: Information control nets: a mathematical model of office information flow. In: P.F. Roth & G.J. Nutt, Proceedings of the ACM Conference on Simulation, Measurement and Modeling of Computer Systems (pp. 225-240). New York: ACM Press, 1979
17. The Eclipse Modeling Framework (EMF), http://www.eclipse.org/emf/.
18. Herbst, J.; Karagiannis, D.: Integrating Machine Learning and Workflow Management to Support Acquisition and Adaptation of Workflow Models. In: Proceedings of the Ninth International Workshop on Database and Expert Systems Applications, IEEE, 1998, pp. 745-752.
19. ITIL ISO/IEC JTC 1/SC/, ISO/IEC TR 15504, 1998
20. IBM Homepage www-306.ibm.com/software/info/developer/index.jsp, access 30 April 2005.
21. Jacobson, I.: Object-oriented Software Engineering. Addison-Wesley, 1992.
22. Jacobson, I.; Booch, G.; Rumbaugh, J.: The Unified Software Development Process. Addison-Wesley, 1998.
23. Kaplan, J.: Auditors Guide to Flowcharting, AUDITNET^{sm,} 2001
24. Kappel, G.; Schrefl, M.: Objektorientierte Informationssysteme: Konzepte, Darstellungsmittel, Methoden. Springer-Verlag, 1996.
25. Keller, G.; Nüttgens, M.; Scheer, A.-W.: Semantische Prozeßmodellierung auf der Basis "Ereignisgesteuerter Prozeßketten (EPK)". Veröffentlichungen des Instituts für Wirtschaftsinformatik, Heft 89, Universität Saarbrücken, 1992.
26. Kühn, H.; Bayer, F.; Junginger, S.; Karagiannis, D.: Enterprise Model Integration. In: Bauknecht, K.; Tjoa, A M.; Quirchmayr, G. (Eds.): Proceedings of the 4th International Conference EC-Web 2003 - Dexa 2003, Prague, Czech Republic, September 2003, LNCS 2738, Springer-Verlag, pp. 379-392.
27. MetaEdit+ Homepage. www.metacase.com, access 30 November 2004.
28. METIS Homepage. www.computas.com, access 30 November 2004.
29. Microsoft Homepage http://office.microsoft.com/en-us/FX010857981033.aspx, access 30 April 2005.
30. ModelMaker tools Homepage www.modelmakertools.com/, access 30 April 2005.
31. Nassi, I.; Shneiderman, B.: Flowchart Techniques for Structured Programming, ACM SIGPLAN Notices, vol 8, no 8, 1973.
32. Object Management Group: Unified Modeling Language Specification, OMG Specification/2003-09-01, September 2003, http://www.omg.org/docs/formal/03-03-01.pdf, access 30 November 2004.

33. Object Management Group: MDA Guide, Version 1.0.1, 12. June 2003. http://www.omg.org/cgi-bin/apps/doc?omg/03-06-01.pdf, access 30 November 2004.
34. Ould, M.; Wiley, A.: Business Processes, Modelling and Analysis for Reengineering and Improvement. John Wiley & Sons, Chichester, 1995.
35. Petri, C.A.: Kommunikation mit Automaten. PhD thesis, Institute for Instrumental Mathematics, Bonn, 1962.
36. Royce, W.: Managing the development of large software sytems: Concepts and techniques. In WESCON Technical Papers, 1970. Reprinted in Proceedings of the Ninth International Conference on Software Engineering, 1987, pp. 328–338.
37. Rumbaugh, J.; Blaha, M.; Premerlani, W.; Eddy F.; Lorensen W.: Object-Oriented Modelling and Design. Prentice-Hall, 1991.
38. Tsichritzis, D.: Form Management, Communications of the ACM, 25(7), 1982, 453 – 478
39. Yourdon, E.: Modern Structured Analysis, Yourdon Press Computing Series, Prentice-Hall, Englewood Cliffs, New Jersey, 1989.
40. Zisman M.: Representation, Specification, and Automation of Office Procedures, PhD Thesis, University of Pennsylvania, 1977.