

A PATTERN-ORIENTED APPROACH TO A METHODOLOGICAL EVALUATION OF MODELING METHODS

Michael Amberg

University of Bamberg, Business Information Systems
D-96045 Bamberg, Tel. (+49 951) 863-2578, Fax (+49 951) 9370412,
e-mail: michael.amberg@sowi.uni-bamberg.de

ABSTRACT

The paper describes a pattern-oriented approach to evaluate modeling methods and to compare various methods with each other from a methodical viewpoint.

A specific set of principles (the patterns) is defined by investigating the notations and the documentation of comparable modeling methods. Each principle helps to examine some parts of the methods from a specific point of view. All principles together lead to an overall picture of the method under examination.

First the core ("method neutral") meaning of each principle is described. Then the methods are examined regarding the principle. Afterwards the method specific interpretations are compared with each other and with the core meaning of the principle. By this procedure, the strengths and weaknesses of modeling methods regarding methodical aspects are identified.

The principles are described uniformly using a *principle description template* according to descriptions of object-oriented design patterns. The approach is demonstrated by evaluating a business process modeling method.

INTRODUCTION

It is commonly accepted that modeling methods assist in system analysis and design. These methods help to develop well-understood high-quality models in an efficient and effective way. The **method based system analysis and design** is embedded in the following overall context (figure 1).

For complex system engineering tasks one uses a formalized interim representation of a real system that is called *model*. The similarities between the model and the real system help to develop a well-engineered solution for the task (e.g. an application system or a reengineering recommendation). The model serves as a communication medium within the development process.

The system engineering task is typically described by one or more **objectives** (such as to develop an enterprisewide application system) and an **object of investigation** (the real system such as the respective enterprise). Then you select an appropriate *modeling method*. The **modeling method** is used to construct a **model** as an abstraction of the real world (object of investigation) considering all relevant views in order to fulfill the prescribed objectives. To meet these requirements a method consists of a set of (semi-formal) **notations and guidelines** about how and when to use the notations.

The example in figure 1 illustrates the overall context. For the *development of an application system* (objective) you might select the *Object Modeling Technique OMT* by Rumbaugh et al. (method). Afterwards you specify an *object model* (the application model) *of the enterprise* (the object of investigation) and use the model to implement the *application system*.

Looking at the available system engineering methods the situation can be described as follows: Numerous methods and various improvements of methods (variants) exist. Additional methods are under development. The variety and the growing number of methods can be explained historically through the expansion and modification

- of the object of investigations (e.g. software, application systems, information systems or business systems),
- of the underlying paradigm (e.g. data-oriented, functional-oriented, object-oriented or business process oriented),
- of the objectives that are pursued (e.g. to develop an application system or to reengineer an enterprise) and
- of the methodical point of views as well as the modeling procedures that are regarded to be suitable.

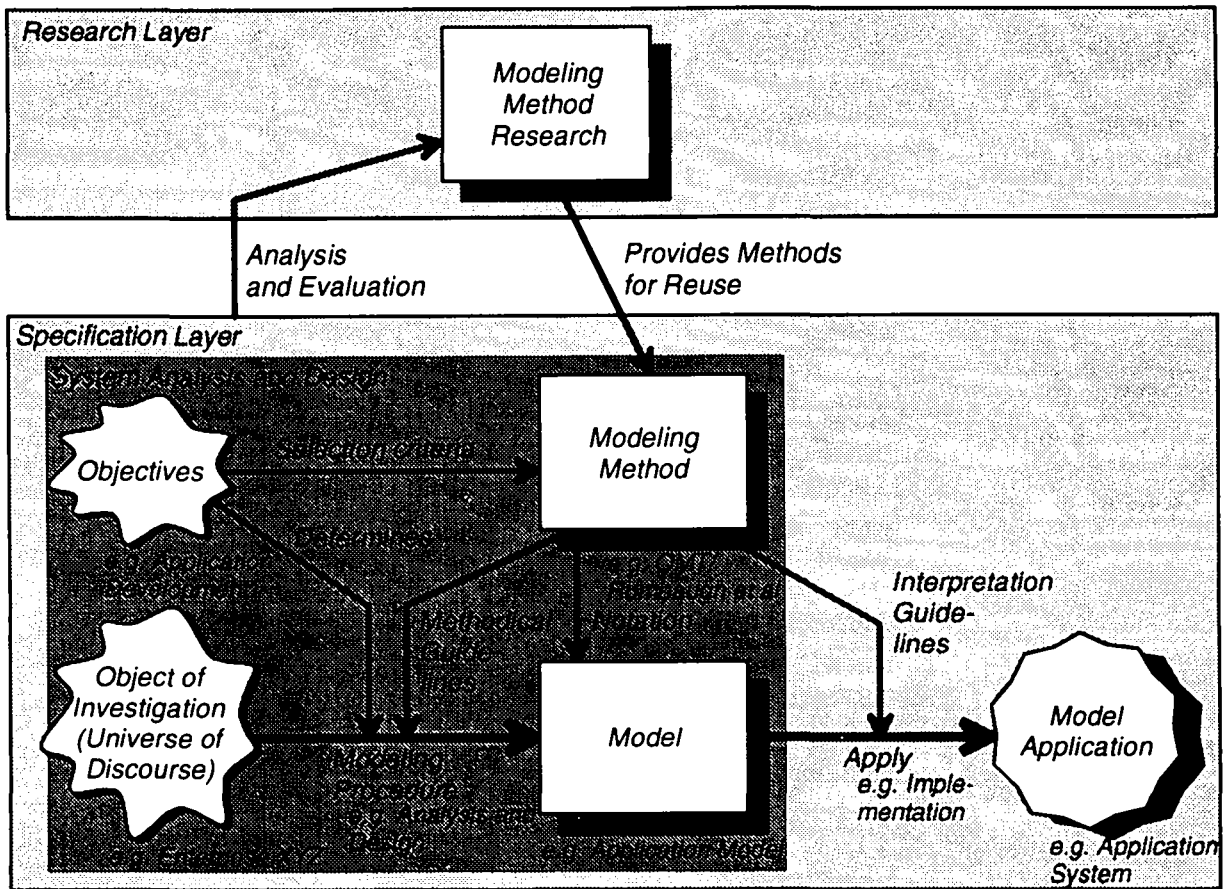


Figure 1: The Overall Context of Method Based System Analysis and Design

Examples of functional-oriented methods are Structured Analysis (SA/De Marco (1979)) and Structured Analysis and Design Technique (SADT/SofTech Ross and Schoman (1977)); of data-oriented methods are Entity Relationship Model (ERM/Chen (1976)) and Structured Entity Relationship Model (SERM/Sinz (1988)); of object-oriented methods are Object Oriented Design (OOD/Booch (1994)) and Object Modeling Technique (OMT/Rumbaugh et al. (1991)); of business process modeling methods are Line of Visibility Engineering Methodology - Enhanced (LOVEM-E/IBM (1995)) and Semantic Object Model (SOM/Ferstl and Sinz (1995)).

A PATTERN-ORIENTED APPROACH TO A METHODOLOGICAL EVALUATION

There are two main approaches apparent for an evaluation of methods: A *methodical evaluation* and an *empirical evaluation*. A **methodical evaluation** analyzes which aspects of real systems are considered by a method and what axioms the method is based on. Whereas an **empirical evaluation** is inductively based on the experiences made when applying methods (e.g. best practice). This paper focuses on a methodical evaluation approach (see for instance Stein (1994)).

Looking at modeling methods the degree of methodical support varies distinctly. For the use of existing methods and for the development of new methods it is important to understand:

- Which methodical viewpoints does a method provide? Which characteristics of the object of investigation are regarded to be relevant?
- Which objectives does a method support? Which guidelines does it recommend?
- Which methodical viewpoints are appropriate for which context?

A methodical evaluation of a method is a difficult task: How to approach the evaluation? Evaluate to which criteria? Is a comparison with other methods appropriate? How should differences be assessed? What aspects are considered in too much detail or not detailed enough? Are relevant aspects missing at all?

To ease the evaluation the paper proposes a **pattern-oriented approach**. A *specific set of principles* (the patterns) is defined by investigating the notations and the documentation of the modeling methods taken into

consideration. Each principle helps to examine some parts of the methods by focusing onto a specific aspect. Thus, the principles guide the evaluation. All principles together lead to an overall picture of the method.

Evaluation procedures

This approach can be used for two main, partly overlapping intentions:

- A. **To become acquainted with a new method and to compare it with other methods:** This is relevant for a method engineer (i.e. a method developer or a user who applies methods) who wants to learn more about an unfamiliar method. Comparing the method with known methods can simplify the understanding. By this, the similarities and the differences are identified regarding methodical aspects. The comparison can be used to **rate** methods concerning given goals and measurements. It can also be used to **rank** methods concerning their methodical support.
- B. **To select a method for a given task:** This is relevant for a user who wants to select a method out of a set of methods and wants to apply the method for a given task. To do this, the user has to determine principles that are relevant for the task. Afterwards the user selects the method that fits the required principles best. Thus the pattern-oriented approach helps to detect the methodical aspects that are addressed or are not addressed by a method. When combining several methods the approach helps to discover the overlapping methodical aspects.

These intentions affect the evaluation procedure. For the first intention an evaluation procedure that produces satisfying results comprises the following steps:

- A1. Determine the method to evaluate and the set of methods to compare it with.
- A2. Detect the **specific set of principles** by investigating the notations and the documentation of the methods. Each principle will be used to examine some part of the methods from a specific point of view.
- A3. Describe the core meaning of each principle by filling out the appropriate slots in the *principle description template* (see 2.2).
- A4. Fill out the slots in the *principle description template* that are related to the method under evaluation.
- A5. Fill out the slots in the *principle description template* that are related to the comparable methods in the sum. Taking all comparable methods into consideration may unveil additional aspects.
- A6. All principles together lead to an overall picture of the method under evaluation. The similarities and differences to the core meaning and to the other methods help to find their strength and weaknesses (problem areas as well as opportunity areas). All principle description templates together are regarded to be an **evaluation schema**. The evaluation schema can directly be visualized by a table (such as in figure 2). The slots of the description template are represented by columns. The context of the principle templates are depicted as rows.

Name	Description (core meaning)	Correlation to Method	Correlation to other Methods	...
Principle 1				
Principle 2				
Principle 3				
...				

Figure 2: An evaluation schema to become acquainted with a new method and to compare it with other methods (case A)

The filled out principle description templates can be reused for the repeated evaluation of modeling methods. For instance if you want to become acquainted with several methods you have to repeat step A4 for every method.

To select a method (case B), the evaluation procedure can comprise the following steps:

- B1. Determine the task to be performed and the set of methods to be compared.
- B2. Detect the **specific set of principles** by investigating the task and the methods.
- B3. Describe the core meaning of each principle by filling out the appropriate slots in the *principle description template* (see 2.2).
- B4. For each method: Fill out the slots in the *principle description template* that are related to the method under evaluation.
- B5. Rate each principle regarding its suitability for the task. Rate each method regarding its support of each principle.
- B6. Compare the required rating of a principle with the provided ratings of the methods for the respective principle. The result of the comparison might be that no, one or several methods provide the methodical support needed. An **evaluation schema** can assist the comparison, visualized by a table (such as in figure 3). A column is specified for the principle's name, for the task and for each method that is examined. The ratings of the principles are depicted as rows.

Name of the Principle	Rating concerning the task	Rating concerning Method 1	Rating concerning Method 2	...
<i>Principle 1</i>	<i>e.g. optional</i>	<i>e.g. ++</i>	<i>e.g. +</i>	
<i>Principle 2</i>	<i>e.g. mandatory</i>	<i>e.g. -</i>	<i>e.g. ++</i>	
<i>Principle 3</i>	<i>e.g. not necas.</i>	<i>e.g. 0</i>	<i>e.g. 0</i>	
...				
<i>Other selection criterias</i>				
...				

Figure 3: An evaluation schema to select one or more methods for a given task (case B)
 Additionally, the evaluation schema can be extended to consider non-methodical selection criteria. Such non-methodical selection criteria are for instance tool support, documentation quality, ease of handling and so on. The problem is to select appropriate measurements and to make the measurements comparable to each other.

A formalism to describe principles in a uniform way

All principles within a system of principles must be described in a uniform way. Such a description should support the evaluation of a method and simplify the comparison of methods. A *description template* is used to describe the principles in a uniform way. Description templates are frequently used to describe object-oriented design patterns. A description template provides a set of slots that are regarded to be useful in the specific context. The principle description template used here is based upon the conceptual ideas introduced by Alexander (1979) and is closely related to the one developed for design patterns in Gamma et al. (1994). Similar to the two description templates it comprises a principle's name, rationale, description and discussion. All other slots are regarded to be specific for the context described here.

The **principle description template** comprises the following slots:

- **NAME.** A name is used to address the principle.
- **RATIONALE.** The motivation (what and why) for the principle is summarized.
- **DESCRIPTION.** The core meaning is described in a method neutral way. The constraints of applying the principle are discussed.
- **METHODOLOGY.** What methodical steps are possible and preferable?
- **CORRELATION TO THE METHOD UNDER EXAMINATION.** How does the method address this principle? What is intentionally not implied?

- **CORRELATION TO OTHER METHODS.** How do comparable methods address this principle? What additional aspects do they imply? What is intentionally not addressed there? Methods are regarded to be comparable, when they support similar objectives, analyze similar objects of investigation and follow the similar paradigm.
- **DISCUSSION.** The relationship to other principles are discussed.

The given set of slots is regarded to be essential. Additional slots may be added if appropriate, e.g. slots for examples, variants, to describe the applicability, to propose improvements or to guide a tool implementation.

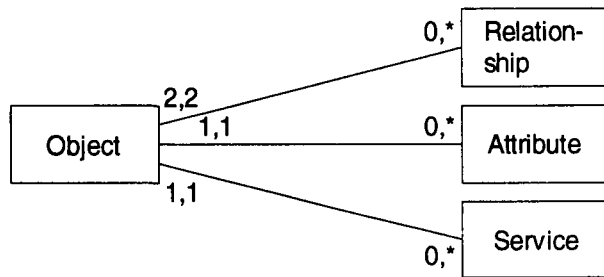


Figure 4: The object model for the description of the methodical coherence regarding a principle

To describe the methodical coherence regarding a principle a graphical notation (an object model, figure 4) similar to a meta-model is used. The **object model** consists of the elementary components *object*, *relationship*, *attribute* and *service*. Therefore the methodical coherence regarding a principle is made out of a set of objects having relationships to each other. Attributes and services are assigned to objects if they are necessary to clarify the semantics. The graphical notation is used to describe the core meaning and the method's viewpoints in detail.

EVALUATION OF A BUSINESS PROCESS MODELING METHOD

The approach was used to evaluate the method LOVEM-E (Line of Visibility Engineering Methodology - Enhanced; IBM (1995)) and to compare it with SOM Ferstl and Sinz (1994, 1995). LOVEM-E and SOM are two alternative methods for business process modeling. They assist business and system professionals in the specification and reengineering of business processes.

The following **specific set of principles** were regarded to be relevant for the evaluation. They were found by investigating the notation (e.g. helping to categorize the terms of the notation; see 3.2) and documentation (e.g. claimed applicability) of LOVEM-E and SOM:

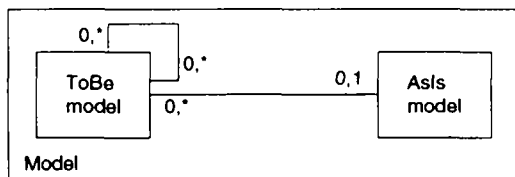
- The **model/view principle** distinguishes between the actual specification results concerning the object of investigation and the presentation of the specification results to users. A model contains the specification results. Views concentrate on the different textual and graphical representations of models. The separation of the functional core from its user representation helps to focus on these aspects separate from each other. Such a separation allows a model to have different visual representations for different users.
- The **AsIs model/ToBe model principle** is used to explicitly specify different models of the object of investigation. Business reengineering is assisted when you specify ToBe models and compare different ToBe models with each other and with an AsIs model.
The **model/requirement principle** is used to explicitly specify the requirements a model has to meet and to separate the requirements from the model specification. The separation helps to specify consistent requirements and helps to verify the consistency of a (core) model concerning the requirements. Measurements that are based on the requirements allow the evaluation of the model. The principle also allows the flexible combination of the requirements with different models. The models are then regarded to be alternative solutions to a given set of requirements.
- The **task/resource principle** distinguishes between what is or what will be done (business task) from how it is or how it will be performed (resource). This differentiation helps to handle complexity by concentrating on either the tasks or the resources of a business as well as on the link between them.
- The **logical model/physical model principle** focuses on abstraction. Abstraction is one of the fundamental principles to cope with complexity. The principle provides explicit barriers between abstractions. In comparison the logical model represents the abstract layer whereas the physical model represents the detail layer.

Example of a filled out principle description template

To demonstrate a filled out principle description template the AsIs model/ToBe model principle is represented (figure 5):

RATIONALE. (already mentioned above)

DESCRIPTION. An **AsIs model** reflects the business at its current architectural level. Any inconsistencies within a model are accepted as long as the inconsistencies also exist in the business. A **ToBe model** is used to show what you would like your business to be. A ToBe model must not be consistent with the real business. But it has to meet the prescribed requirements (see the model/requirement principle).



You normally have a single AsIs model and none, one or more ToBe models concerning a single object of investigation. Different ToBe models can be linked to each other. You can also specify ToBe models without explicitly specifying an AsIs model.

METHODOLOGY. You can either specify AsIs models, ToBe models or both. The standalone specification of an AsIs model can be used to document your business. The standalone specification of a ToBe model can be used to present a vision or to improve your business. You can for instance use a given reference model to customize a ToBe model for your enterprise. It is generally acceptable and easier to start with the explicit specification of an AsIs model. You have to overcome a smaller abstraction level and you learn to understand the actual business as well as the naming conventions. Afterwards one or more ToBe models can be specified. You can then relate a ToBe model to the AsIs model (either explicitly or implicitly) or to other ToBe models (e.g. comparing alternative solutions).

CORRELATION TO LOVEM-E. LOVEM-E does only rudimentary support this principle by now.

Different models can be named as AsIs models or ToBe models. Some elementary design guidelines are given. The comparison of models is not further addressed. By now no special transformation functions are offered to support a step-by-step transformation of an AsIs model into a ToBe model. Also the comparison of two models is not supported, for instance to identify the location of differences and similarities within two models or to evaluate different models.

CORRELATION TO SOM. SOM has started to support this principle (see Ferstl and Sinz (1995)). The strongly hierarchical approach using different layers of abstraction and refinement simplifies the identification of relevant AsIs and ToBe differences.

DISCUSSION. There are different implementations of the principle possible using the model/view principle. Either an overall model consists of a set of sub-models for each AsIs and ToBe model. Different sub-model can then be closely interrelated. Or the overall model reflects exactly one AsIs or ToBe model. Then the models are far more independent.

The first solution allows to define separate views for each sub-model as well as views on the relationship of different sub-models. The second solution allows primarily the definition of separate views for each model. To provide a relationship among different models naming conventions must be used (e.g. instances of different models with the same name are identical instances). This principle may be expanded in the long run to cope with the evolution of a business. In course of time you will have a set of AsIs models and sets of alternative ToBe models. The analysis of historical AsIs models in comparison to the performed business may then give you hints for improving your business.

Figure 5: An example for an filled out principle description template (AsIs model/ToBe model principle)

Categorization of the notation according to the principles

The notation of LOVEM-E is very extensive. It comprises a rich set of terms (more than a hundred terms). As a result of the evaluation, the principles assisted in categorizing the terms of LOVEM-E according to the

principles. A term was assigned to a categorization if it contributes to the respective principle. Figure 6 illustrates such a categorization for some essential terms of LOVEM-E.

Principle	Categorization	
Model/view principle	Model: <i>(see model/requirement principle)</i>	View: <i>(Charts:)</i> ALOVC, HSD, LLOVC, PLOVC, LTL <i>(Chart elements:)</i> Customer band, Logical band, Manual/Automation line, Manual band ...
AsIs model/ToBe model principle	AsIs model: <i>(no specific terms)</i>	ToBe model: <i>(no specific terms)</i>
Model/requirement principle	Model: <i>(see logical model/physical model principle)</i>	Requirement: Assumption/Issue/Recommendation; Critical Measurement Point; Critical Success Factor; Goal/Strategy/Policy; Opportunity Area; Problem Area
Task/resource principle	Task: Activity Customer activity Customer process Data bus Process System Task Data flow	Resource: Business branch Business function Customer band External organization Int/ext organization unit Logical band Organization unit
Logical model/physical model principle	Logical model: Business branch Business function Customer band Customer process Data flow External organization LLOVC Logical band Organization unit Process Process bus	Physical model: Activity Customer activity Int/ext organization unit JLOVC PLOVC System Task

Figure 6: Simplified categorization of the notation of LOVEM-E according to the principles

SUMMARY

This pattern-oriented approach assists people (from a methodical point of view) in becoming acquainted with a new method, in comparing methods with each other and in selecting one or more methods out of a given set of methods for a prescribed task.

A flexible evaluation schema is defined for the methodical evaluation of a modeling method by a specific set of principles (the patterns). Each principle helps to examine and evaluate some parts of the method by focusing onto a specific aspect. All principles together lead to an overall picture of the method.

The principles help to guide an evaluation. They point out the context where methods provide good, poor or no support. By this, the areas are discovered for which a modeling method is appropriate.

ACKNOWLEDGMENT

This document is a result of a research visit taken place during 8/7/95 and 9/15/95 at the GSDL (German Software Development Laboratory) of the IBM Deutschland Entwicklung GmbH in Boeblingen. In this time the outlined approach was used to evaluate a business process modeling method under development and to compare it with another process modeling method.

REFERENCES

- Alexander C. (1979) **The Timeless Way of Building**, Oxford: University Press
- Booch G. (1994) **Object Oriented Analysis and Design with Applications**, 2nd edition, Massachusetts: Addison Wesley
- Chen P. (1976) **The Entity Relationship Model - Towards a Unified View of Data**, ACM Transaction on Database Systems, Vol. 1, No. 1, pp 9-36
- DeMarco T. (1979) **Structured Analysis and System Specification**, Englewood Cliffs, New Jersey: Prentice Hall
- Ferstl O.K., Sinz E.J., Amberg M., Hagemann U., Malischewski C. (1994) **Tool-Based Business Process Modeling Using the SOM Approach**, Bamberger Beiträge der Wirtschaftsinformatik Nr. 19
- Ferstl O.K., Sinz E.J. (1994) **Multi-layered Development of Business Process Models and Distributed Business Application Systems**, Bamberger Beiträge der Wirtschaftsinformatik Nr. 20
- Ferstl O.K., Sinz E.J. (1995) **Re-Engineering von Geschäftsprozessen auf der Grundlage des SOM-Ansatzes**, Bamberger Beiträge der Wirtschaftsinformatik Nr. 26
- Gamma E., Helm R., Johnson R., Vlissides J. (1994) **Design Patterns - Elements of Reusable Object-Oriented Software**, Massachusetts: Addison-Wesley
- International Business Machines Corp. IBM (1995) **Business Modeling Tool - User's Guide**, Revised First Draft, Release 1
- International Business Machines Corp. IBM (1995) **IBM LOVEM-E - Interim User's Guide**, Draft, Version 3
- Ross D.T., Schoman K.E. (1977) **Structured Analysis for Requirements Definition**, Trans. On Software Engineering, Vol. SE-3, No. 1, pp 6-15
- Rumbaugh J., Blaha M., Premerlani W., Eddy F., Lorensen W. (1991) **Object-Oriented Modeling and Design**, Englewood Cliffs, New Jersey: Prentice Hall
- Sinz E.J. (1988) **Das Strukturierte Entity-Relationship-Modell (SER-Modell)**, Angewandte Informatik 5, pp 191-202
- Stein W. (1994) **Objektorientierte Analysemethoden - Vergleich, Bewertung, Auswahl**, Mannheim: BI Wissenschaftsverlag