

On the Evolution of Methods for Conceptual Information Systems Modeling



Elmar J. Sinz

1 Conceptual Modeling

Models are among the most fundamental utilities of science. Becker and Vossen express their understanding of a model as follows: A model is an immaterial and abstract image of the real world for the purposes of a subject (Becker & Vossen, 1996, p. 19). It is abstract because it reduces the real world to the essential and omits unnecessary details. The purpose of a model is to capture the real world in terms that enable the subjects to understand, verify, communicate, build and apply the model. Finally, the subject of a model covers all its stakeholders.

Conceptual modeling is a special form of modeling. Thalheim underlines three dimensions: (1) Modeling language constructs that need to be well understood in syntax, semantics and pragmatics; (2) collection of application areas that allow understanding the problems to be solved; (3) engineering that allows reducing the problems to a manageable size (Thalheim, 2010, Sect. 1.1). Strahringer, who is focused on business problems, says that conceptual modeling is the creation of information models at the level of a business concept, which are strongly oriented towards the business problem and the common technical language and abstract from the IT-technical implementation (Strahringer, 2013).

This means that a conceptual model is specified in terms of the real world. The modeling method helps to capture the real world and to specify it as a model.

E. J. Sinz (✉)
University of Bamberg, Bamberg, Germany
e-mail: elmar.sinz@uni-bamberg.de

2 Conceptual Modeling of Information Systems

One of the fundamental concepts of organizational theory is the term task. Every goal-oriented human action takes place in tasks. Tasks are generally performed by humans. With the development of business information systems engineering, a second actor stepped up to the human: the machine.

In the following we will only look at the tasks of information processing. These tasks, whether they are human or machine tasks, establish the information system.

Subsequently, a framework contributed by organizational theory is used (Fig. 1). It is based on the concept of the task, described from an inside and an outside perspective (Ferstl & Sinz, 2013, p. 95 ff).

The outside perspective of a task (grey) comprises a task object (O), goals (G) and pre- and post-events (E). The latter are prerequisites for the execution of a task or result from the execution. The goals specify the desired states of the task object after the task has been executed.

O, G and E determine the views on the outside perspective. The three views together allow a complete description of the outside perspective. If one or two views are missing, the description is correspondingly incomplete.

The inside perspective (white) refers to the solution procedure of the task. It consists of the actions (F) that are executed on the task object (D, I). The interaction component (I) assumes that the interaction channels between the sub-objects are part of the task object. The action control (P) triggers the respective actions; the actions report their results to the action control. The action control pursues the goals given in the outside perspective; it is released by pre-events and produces post-events.

D, I, F determine the static views, P is the dynamic view of the inside perspective of a task. The views of the inside perspective are terminologically closer to the IT-technical implementation. As in the outside perspective, the more views are given, the more complete is the specification of the inside perspective.

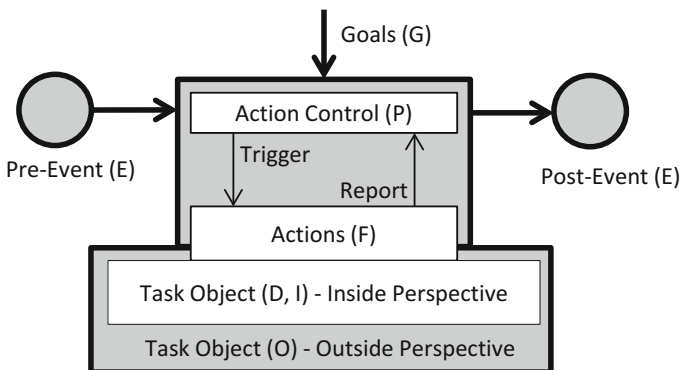


Fig. 1 Organizational view on a task (Ferstl & Sinz, 2013, 95ff)

To give an example, the order entry of a sales company is used. The task object consists of customers, articles and orders. The goal is to have a new order entered to the task object after the task is performed. The pre-event is marked by the arrival of an order that starts the solution procedure for the task. For this purpose, the action control triggers actions to check the customer, to check articles, to reserve items and finally write the entered order to the task object. The task is completed with the creation of a post-event for the entered order. This could be the pre-event for order picking, for example.

3 Evolution of Conceptual Information Systems Modeling

This short paper traces the evolution of important methods for conceptual information systems modeling.¹ Like many other trends, the evolution floats upstream, i.e. it goes step by step from inside to outside perspective. It starts with a look at programming (inside perspective) and ends with a complete conceptual modeling (outside perspective). Looking at the modeling methods, evolution begins with functional decomposition and ends with event-driven modeling.

As evolution progresses, more complex systems can be modeled. The complexity manifests itself the degree of automation and integration that can be managed via the models.

The following methods for information systems modeling have gained relevance and visibility in theory and practice (extended from Ferstl and Sinz 2013, p. 139 ff). All the methods can be applied to conceptual modeling, especially because their terms bridge the gap to the real world (Table 1).

3.1 *Functional Decomposition*

Functional decomposition is one the first methods used for conceptual modeling. It looks on the world as a function to be decomposed. The decomposition leads to sub-functions and so on until the sub-functions are small enough to be programmed. The method is limited to the inside perspective and the function view (F). Data (D) only occurs in the leaf nodes of the sub-function tree, but it is referenced and not modeled. The outside perspective does not occur.

Nowadays the functional decomposition plays no role within conceptual modeling. An example of a method is HIPO (Hierarchy of input-process-output) (e.g. Balzert, 1982).

¹The evolution of modeling methods is presented from the subjective perspective of the author.

Table 1 Information systems modeling methods

Modeling method	Real world view	Example	Perspective	View
Functional decomposition	The world is a function to be decomposed	HIPO	Inside	F
Data flow modeling	The world is a set of data flows, transformed by activities	SADT, Structured Analysis	Inside	I, F, D
Data modeling	The base of the world is a set of related data structures	ERM, SERM, RM	Inside	D
Object modeling	The world consists of a set of interacting objects, each encapsulating data and function	OOA, OMT	Inside Outside	D, F, I O
Process modeling	The world is a set of processes	EPC, BPMN	Outside Inside	E P, F
Object and process modeling	The world consists of objects, which are coordinated to configure business processes	SOM	Outside Inside	G, O, E D, F, I, P
Service-oriented modeling	The world is a system of delivering and consuming services	e.g. Erl	Outside	G, O, E
Event-driven modeling	The world is a system of components, listening for events	e.g. Rommelspacher	Outside	G, O, E

Legend Inside perspective with static views on (F) function, (D) data, (I) interaction and a dynamic view on (P) processes. Outside view on (G) goals, (O) task object, (E) events (see Fig. 1)

3.2 Data Flow Modeling

Data flow modeling has accompanied the practical systems engineering for several decades. It looks at the world as a set of data flows, transformed by activities. Starting with the main activity, the activities are decomposed successively. At each layer, the data flows are modeled, so that a complete model is available along the hierarchy of decompositions. The method comprises three views, mainly the interaction (I) of activities, the activities themselves (F) and the data flows (D). Interestingly, the data within the data flows is referenced, but not modeled.

The method is used for the inside perspective because there is no notion for task object, goals or events. The main shortcoming of the method is its focus on activities as the item to be decomposed.

An example for a data flow modeling method is SADT (Structured Analysis and Design Technique) (Balzert, 1982). Another well-known example is SA (Structured Analysis) (DeMarco, 1979; Gane & Sarson, 1979) which is available in many of the older CASE tools.

3.3 Data Modeling

Data modeling considers the world as a set of related data structures. This is the platform on which all the functions can be implemented. Data models usually have a certain level of detail, hierarchical decomposition is not possible. The method focuses only on data (D). In contrast to the methods mentioned above, data structures are modeled with attributes, keys and relationships.

Data modeling is the first method used for enterprise-wide models. Its major disadvantage is the global view on data, which prevents a meaningful modularization of the overall system. Moreover, it is restricted to the inside perspective.

Examples of methods are the RM (Relational Model) (Codd, 1970), which comes closer to implementation, the ERM (Entity-Relationship Model) (Chen, 1976) or its modification for large schemas, the SERM (Structured Entity-Relationship Model) (Sinz, 1988).

3.4 Object Modeling

Object modeling marks a major step in programming, design and modeling of information systems. It looks at the world, which consists of a set of interacting objects, each encapsulating data and function. From programming languages like Smalltalk-80, and later C++ and Java, the principles for object modeling were adopted. OOA (Object-Oriented Analysis) (Coad & Yourdon, 1991) and OMT (Object-Modeling Technique) (Rumbaugh, Blaha, Premerlani, Eddy, & Lorensen, 1991) are among the first object modeling methods.

The success of object modeling is the concept of class, which encapsulates data (D) and function (F) without telling the environment how they are implemented. In addition, the interaction channels between classes (I) are defined.

Object modeling combines the three views D, F and I and thus enables an almost complete inside perspective. It is also used to model an outside perspective, despite missing goals and events.

3.5 *Process Modeling*

Process modeling looks at the world as a set of processes. It is the first time that a new view of modeling becomes visible: the process view (P). Process modeling is a natural continuation of object modeling. It sequences the small pieces of operations on objects.

Applied to the outside perspective of a task, process modeling is put into practice by event-driven process chains (EPC) (Nüttgens, 2013). The focus here is on events (E); the fact that functions are modeled and not task objects is ignored.

Used for the inside perspective, process modeling concentrates on the process view (P), describing the sequence of functions (F), which are not modeled in connection.

Another method for process modeling is business process model and notation (BPMN) (OMG, 2011). Here, switching between inside and outside perspective is supported by white-box and black-box pools. The interaction between the pools is described by message flows.

Process modeling attracted attention with business process modeling.

3.6 *Object and Process Modeling*

The world consists of objects, which are coordinated to configure business processes. Thus, this method combines object and process modeling.

This class of modeling methods is represented by the semantic object model (SOM) (Ferstl & Sinz, 1995). On the business process layer of the enterprise architecture, the outside perspective of tasks is modeled with task object (O), goals (G) and events (E). The tasks are connected by transactions that describe the coordination of the tasks. Two principles of coordination are available: choreography (negotiation; market) and orchestration (feedback-control loop; hierarchy).

The corresponding inside perspective is represented by the resource layer. The solution procedure of each task is modeled by conceptual objects representing the task objects (D) and the actions (F) on them. The interaction channels (I) are derived from the transactions. Process objects represent the action control (P).

3.7 *Service-Oriented Modeling*

In service-oriented modeling, the world is seen as a system of delivering and consuming services. It emphasizes the outside perspective on tasks and offers a lot of solutions for the inside perspective. On the one hand the notion of service is very close to the real world and on the other hand it harmonizes with implementation techniques, like the client-server principle. Therefore, service-oriented modeling is currently very popular.

Erl (2007) gives examples for service-oriented modeling methods.

3.8 Event-Driven Modeling

The world is seen as set of components, e.g. classes or services that listen to events (Rommelspacher, 2008). When a component feels responsible for an event, it starts to process it. No interaction channel must be predefined. Complex events are supported, i.e. compound events consisting of simpler events. As in service-oriented modeling, it emphasizes the outside perspective of tasks and enables a lot of solutions for the inside perspective.

Event-driven modeling convinces with its flexibility. If one component fails, another (if available) can take over the task. One problem with the inside perspective is the infrastructure that is necessary for event handling.

4 Lessons Learned

This paper takes an alternative view of modeling methods suitable for conceptual modeling. It considers the outside and inside perspective of tasks, the organizational building block of information systems. From this point of view, it is examined whether a modeling method can contribute to conceptual modeling.

If we look at the development of the methods, we see that evolution floats upstream from a single view on the inside perspective to multiple views and to the outside perspective. Modeling the outside perspective unfolds the full potential of conceptual modeling. Otherwise, starting with the outside perspective allows a secured transition to the inside perspective.

The evolution of methods for conceptual information systems modeling not only enables an increasing degree of automation and integration of information systems. At the same time it is also due to business-IT alignment, which tries to flexibly adapt IT to business requirements and to adopt upcoming IT platforms and technology to new business models.

References

- Balzert, H. (1982). *Die Entwicklung von Software-Systemen. Prinzipien, Methoden, Sprachen, Werkzeuge*. Mannheim: B.I.-Wissenschaftsverlag.
- Becker, J., & Vossen, G. (1996). Geschäftsprozeßmodellierung und Workflow-management: Eine Einführung. *Geschäftsprozeßmodellierung und Workflow-management—Modelle, Methoden, Werkzeuge* (pp. 17–26). Bonn: International Thomson Publishing.
- Chen, P. P.-S. (1976). The Entity-Relationship model: Toward a unified view of data. *ACM Transactions on Database Systems*, 1(1), 9–36.
- Coad, P., & Yourdon, E. (1991). *Object-oriented analysis* (2nd ed.). Englewood Cliffs, NJ: Yourdon Press.
- Codd, E. F. (1970). A relational model for large shared data banks. *Communications of the ACM*, 13(6), 377–387.

- DeMarco, T. (1979). *Structured analysis and system specification*. Englewood Cliffs, NY: Yourdon Press.
- Erl, T. (2007). *SOA: Principles of service design* (1st ed.). Upper Saddle River, NJ: Prentice Hall.
- Ferstl, O. K., & Sinz, E. J. (1995). Der Ansatz des Semantischen Objektmodells (SOM) zur Modellierung von Geschäftsprozessen. *Wirtschaftsinformatik*, 37(3), 211–220.
- Ferstl, O. K., & Sinz, E. J. (2013). *Grundlagen der Wirtschaftsinformatik* (7th ed.). München, Deutschland: De Gruyter Oldenbourg.
- Gane, C., & Sarson, T. (1979). *Structured systems analysis: Tools and techniques*. Englewood Cliffs, NJ: Prentice Hall.
- Nüttgens, M. (2013). EPK. Retrieved September 23, 2018, from <http://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/is-management/Systementwicklung/Hauptaktivitaten-der-Systementwicklung/Problemanalyse-/Geschäftsprozessmodellierung/EPK/index.html>.
- OMG. (2011). About the business process model and notation specification version 2.0. Retrieved September 23, 2018, from <https://www.omg.org/spec/BPMN/2.0>.
- Rommelspacher, J. (2008). Ereignisgetriebene Architekturen. *Wirtschaftsinformatik*, 50(4), 314–317.
- Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., & Lorensen, W. (1991). *Object-oriented modeling and design*. Upper Saddle River, NJ: Prentice Hall.
- Sinz, E. J. (1988). Das Strukturierte Entity-Relationship-Modell (SER-Modell). *Angewandte Informatik*, 5, 191–202.
- Strahinger, S. (2013). Konzeptuelle Modellierung von IS—Enzyklopaedie der Wirtschaftsinformatik. Retrieved September 23, 2018, from <http://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/is-management/Systementwicklung/Hauptaktivitaten-der-Systementwicklung/Problemanalyse-/konzeptuelle-modellierung-von-is>.
- Thalheim, B. (2010). Towards a theory of conceptual modelling. *Journal of Universal Computer Science*, 16(20), 3102–3137.



Elmar J. Sinz Prof. Dr. Elmar J. Sinz, born 1951 in Straubing, received a diploma (FH) in mechanical engineering in 1972 and a diploma in business administration in 1977. He completed his doctorate (Dr. rer. pol.) in 1983 with the thesis “Construction of Business Information Systems” at the University of Regensburg and his habilitation (Dr. rer. pol. habil.) at the same university in 1987 with the thesis “Data Modeling of Business Problems”. After deputising a professorship at the University of Marburg in 1987-1988, he joined the University of Bamberg in 1988, where he became a full professor for information systems, in particular systems engineering and database application.