

# Navigating the Methodology Jungle –

The communicative role of modelling techniques in information system development

S.J.B.A. Hoppenbrouwers

University of Nijmegen  
Sub-faculty of Informatics  
Iris Research Group  
Toernooiveld 1  
6525 ED Nijmegen  
The Netherlands, EU  
S.Hoppenbrouwers@cs.ru.nl

H.A. Proper

University of Nijmegen  
Sub-faculty of Informatics  
Iris Research Group  
Toernooiveld 1  
6525 ED Nijmegen  
The Netherlands, EU  
E.Proper@cs.ru.nl

V.E. van Reijswoud

Uganda Martyrs University – Nkozi  
Department of Computer Science and  
Information Systems  
PO Box 5498 Kampala  
Uganda  
Victor@umu.ac.ug

September 20, 2005

PUBLISHED AS:

S.J.B.A. Hoppenbrouwers, H.A. (Erik) Proper, and V.E. van Reijswoud. Navigating the Methodology Jungle – The communicative role of modelling techniques in information system development. *Computing Letters*, 1(3), 2005.

## 1 Introduction

In this position paper, we claim that more attention should be paid to the *communicative role* of modelling techniques in information system development. The communicative role of a modelling technique refers to it providing a *language* for communication between the different *actors* involved in system development, about particular aspects of the system being developed.

When using the term *modelling technique* we (roughly) refer to the combination of (1) a modelling language/notation and (2) procedures/guidelines for the creation of models. This use of the term *modelling techniques*, is in line with definitions that can be found in e.g. [OHM<sup>+</sup>88, Avi95, BMS98]. Our understanding of *model* (and its *model description*) is rather broad. In line with [FVV<sup>+</sup>98], we consider a model to be a purposely abstracted, clear, precise and unambiguous conception of some domain. Given a modelling technique, a *model description* (or representation) can be produced using the syntactic constructs of the modelling language provided by the technique, while following its modelling procedures/guidelines. The underlying modelling language could be a formal language (i.e. a language with a well-defined semantics) or an informal language. It could have a one-dimensional representation style (e.g. natural language, mathematical languages, etc.), or a multi-dimensional representation style (e.g. a graphical modelling language, an animation based language, etc.)

During the development of an information system, practitioners quite often select modelling techniques rather ad hoc, without explicit reasoning as to the suitability of the technique to the (communicative) tasks at hand. As a result, “accidents” in the selection of these techniques do indeed occur. For example, a modelling technique which is well suited to the modelling and communication of functionality and structure of software (e.g. UML) is not likely to be suitable to communicate the impact which future system may have on the business processes using it, to business management. This may sound logical and maybe even trivial; nevertheless, selection of techniques that are inapt for the task at hand does occur in practice [BPH04].

Meanwhile, scholars have been produced numerous modelling techniques [Bub86, AW91, Avi95, BMS98], adding to the multitude of modelling techniques practitioners can select from. The

authors of this paper have themselves contributed their fair share of modelling techniques [BHW91, HW93, BBMP95, PW94, CP96, CHP96, HVH97]. The resulting plethora of techniques has, in the past, already been referred to as “a methodology jungle” [Avi95]. This jungle leaves developers of information systems with the burden of selecting modelling techniques that are apt for the modelling/communication tasks at hand. Quite often, these modelling techniques are explicitly based on the (communicative) requirements posed on them as a result of the roles they are destined to play in information system development.

The selection of modelling techniques from the “methodology jungle” has been addressed before by other scholars. In [HW92, HP98, HR00], it was argued how the “methodology jungle” may be chopped down by formalising both syntax and semantics of the modelling languages underlying these techniques. This observation has inspired us, in our former work on modelling techniques, to at least provide these techniques [BHW91, HW93, BBMP95, PW94, CP96, CHP96] with a formal underpinning. Even though a formalisation of syntax and semantics may indeed lead to some clearings in the jungle, it does not provide insight into the utility [PVH05] of a modelling language with regards to a modelling task at hand. Developers are still left with the task of finding an apt modelling technique to fit the goals of their modelling task.

The field of *method engineering* [Bri96, RB96], does aim to provide a better rationalisation of the selection of so-called *method fragments* that are suitable to a situation at hand, where a (part of a) modelling technique should be regarded as a method fragment. This has resulted in complex and layered modeling approaches to analyse and model organisations and information systems. Even though method engineering does provide guidelines [Bri96] for the selection of method fragments (such as modelling techniques), it does not provide a deep study into the role of modelling techniques as a means of communication.

In this paper, we hypothesise that more attention needs to be paid to the *communicative role* which modelling techniques play during information system development, and furthermore, that criteria for selection need to be developed. Having such an understanding, would contribute towards improved selection (and construction) of modelling techniques for specific (communicative) tasks during information system development. In the remainder of this position paper, we undertake a first exploration in building up a fundamental understanding of this communicative role. In doing so, we take the view that information system development can be regarded as a communication-driven knowledge transformation process, and that a modelling technique’s main purpose is to provide a means (language) for communication (sharing of knowledge).

What we present in our exploration is a (still loosely coupled) conceptual framework to reason about the role of modelling techniques as a communication means in information system development. In figure 1 we depict the conceptual framework as it stands at present<sup>1</sup>. Work on refinement, integration, instantiation, and *validation* of this framework is underway, but we would very much welcome additional efforts to explore and further this field of research.

## 2 Development Community

Given a focus on communication, it is important to identify the elements that can play a role in the communication taking place during system development. We will make a distinction between elements that can be regarded as doing the actual *communicating* and elements that can be regarded as being *communicated upon*. The former class of elements will be referred to as *actors*, while the latter class is referred to as *representations*.

The (human) actors in a system development community, are likely to have some stake with regards to the system being developed. Examples of such actors are: problem owners, prospective actors in the future system (such as the future ‘users’ of the system), domain experts, sponsors, architects, engineers, business analysts, etc. The other class of elements, the *representations*, comprises the many different documents, models, forms, etc., that *represent* bits and pieces of

---

<sup>1</sup>This model has been represented using a refined version [CP96, Pro99] of Object-Role Modelling [Hal01].

knowledge pertaining to the system being developed.

A system development community harbours knowledge about the system being developed. To be more precise, elements in the system development community can be regarded as *knowledge carriers* harbouring knowledge pertaining to (their view on) a sub-domain within the system being developed (and/or its development process). In this vein, the communication occurring within a system development community essentially aims to create, further, and disseminate this knowledge. The actual knowledge can concern the system being developed, as well as the development process itself. In section 5, we will provide a more elaborate discussion on the kinds of knowledge that may (have to) be communicated.

The *actors* in a system development community will (typically as a consequence of their personal *goals* and *stakes*) have some specific interests with regards to the system being developed. Depending on the concerns of a stakeholder, she will be interested in different knowledge topics related to the system being developed. For example: a financial controller will be interested in an investment perspective on the overall scope of a future system, a designer will be interested in all aspects of the design chain from different perspectives, etc.

### 3 Communication-driven

Key to our view on the selection of modelling techniques is their role as a means of communication in system development. In the past we have already taken a communication-driven perspective on modelling activities in information system development [DFW96, HVH97, RMD99, FW04, FW04, BPH04, PH04], as well as the act of system development itself [VHP04]. We are certainly not alone in doing so [NH89, EKW92, Hal01].

To better understand the role of modelling techniques in system development, we first focus on *creation* and *sharing* of knowledge in the development process. In essence, we regard system development as a communication-driven *knowledge transformation* process whereby conversations are used to share and create knowledge pertaining to both the system being developed, as well as to the development process itself. The notion of conversation should be interpreted here as ranging from a single person producing a model (description), via a one-on-one design/elicitation session, to a workshop with several stakeholders and even the widespread dissemination of definitive system designs. We do *not* claim that viewing information system development as a knowledge transformation process is new [Myl98]. Our aim is to use this perspective on system development to better understand and articulate the *requirements* that (should) underly modelling techniques. From this perspective, modelling techniques should be regarded as a means (a language) to an end (system development), in line with the functional perspective on language [Cru00]: (*what is it to be used for?*)

In communication theory, one commonly identifies three layers in human to human communication [Aus62, Sea69, Sea79, Hab84, Sea95]: *syntaxis* (describing the structure of communication), *semantics* (describing the meaning of communication) and *pragmatics* (describing the social impact of communication). These three layers are always present in communication but can be distinguished and studied separately. When the three layers are considered in the context of information system development, we hypothesize that the actors involved emphasise one of the three communication layers depending on their role or stake in the result of the process. If the actor concentrates on the role of the information system for an organization (e.g. management or business consultant) they are more likely to emphasize the social impact of the system (the pragmatical aspect) than the structure of the information entities in the organization. On the other hand, a database designer will concentrate more on the information entities and its attributes (syntaxis) when communicating with the organizational actors and leave the business value of the information outside the scope. The decision to emphasise one of the communication layers is not a deliberate choice, but forced upon the actor by the responsibilities in the information systems development process. We call this the actor-communication focus in the information systems

development process.

To further understand how the actor-communication focus can be used we will relate it to the system development process. An information system development process involves more than just the traditional analysis and design phases, even though many of the available modelling techniques are targeted towards these (early) phases in the development process. Knowledge transformations take place in all the phases in the information system development process. Consider, for instance, the phases as identified in the Rational Unified Process (RUP) [Kru00]: *inception*, *elaboration*, *construction* and *transition*. In each of these phases, knowledge transformations take place. In the initial stages, the focus will be on elicitation and validation of requirements as well as on construction and validation of the system’s design, while in the later stages, the focus will be more on shaping and preparing for the actual implementation of the system in its usage environment. One can readily observe how during system development many different classes of activities (disciplines) are involved, and that from phase to phase the focus on these classes will shift: from requirements and design to change management and deployment. In RUP, the following core disciplines are identified: *business modelling*, *requirements*, *analysis and design*, *implementation*, *test*, *deployment* (in addition to the management of configurations, change, project and environment). Given the three layers of communication as discussed above, one could relate the core disciplines to these layers, in an attempt to identify the dominant form of communication. This leads to the following (tentative) table:

	business modelling	requirements	analysis & design	implementation	testing	deployment
Pragmatics	++	+			+	++
Semantics	+	++	++	+	++	+
Syntaxis			+	++	+	

where the number of + symbols signifies the relative importance of the layer of communication.

## 4 Knowledge Transformations

During the development of a system the knowledge about the system and its development, as it is harboured by a development community, will evolve. New insights emerge, designs are created, views are shared, opinions are formed, design decisions made, etc. Consequently, the knowledge as it is present in a development community can be seen to evolve through a number of *states*. Knowledge needs to be *introduced* into the community first, either by creating the knowledge internally or importing it from outside of the community. Once the knowledge has been introduced to a community, it can be *shared* among different knowledge carriers. Sharing knowledge between different actors may progress through a number of stages. We distinguish three major stages:

**Aware** – An actor may become aware of (possible) knowledge by way of the sharing by another actor (possibly from outside the community), or by creating it themselves.

**Agreed** – When shared, an actor can make up her own mind about the shared knowledge, and decide whether or not to *agree* to the knowledge shared.

**Committed** – Actors who agree to a specific knowledge topic may decide to actually commit to this knowledge. In other words, they may decide to adopt their future behaviour in accordance to this knowledge.

There is no way to objectively and absolutely determine the level of awareness, agreement, or commitment of a given set of knowledge carriers. It is in the eyes of the beholder. This “beholder”, however, will typically be an actor in the system development community. We can, therefore, safely presume that some actors in the system development community will be able to (and have a reason to) judge the level of sharing of knowledge between sets of actors, and communicate about this.

## 5 System Development Knowledge

Thus far, we have not yet considered the knowledge topics about which is communicated during system development. The aim of this section is to explore, at a high level of abstraction, the possible topics of communication.

A first distinction can be made between:

**Target domain** – Knowledge pertaining to the *system* being developed.

**Project domain** – Knowledge about the *development process* that brings forth the system.

We have borrowed the terms *target domain* and *project domain* from the Information Services Procurement Library (ISPL) [FV99a]. For both of these knowledge domains, further refinements can be made with regards to the possible topics. The following additional characterizations can be distinguished:

**Perspective** – Artifacts, such as systems, can be considered from different perspectives. Some examples are: (1) Business, application, and infrastructure aspects of a (computerized) information system; (2) Social, symbolical, and physical aspects of a system; (3) Process, information, actors, and technology featuring in a system.

**Scope** – Given a domain, such as a system or a development project, several scopes can be identified when approaching the domain. Some examples are: (1) enterprise wide; (2) department specific; (3) task specific.

**Design chain** – When considering the design of some artifact, a distinction can be made between: (1) the **purpose** for which an artifact is needed; (2) the **functionality** which the artifact should provide to its environment; (3) the **design** of the artifact, i.e. *how* it should realize the functionality; (4) the **quality** of the artifact, i.e. *how well* it should do so; (5) the **cost** at which it will/may do so, and at which it may be constructed.

**Historical perspective** – Given an artefact with a design, one may consider different versions of this design over time. One could, for example, make a distinction between a strategic (5-10 years), a tactical (1-5 years), and an operational perspective (now).

**Abstraction** – When considering a domain, one may do so at many different levels of abstraction. Also, various forms of abstraction can be distinguished, for example type-instance, generalisation, is-a, encapsulation, and the hiding/encapsulation of implementation details.

**Communication level** – The layers or aspects (syntaxis, semantics, pragmatics) of communication that are considered central. See section 3.

In general, each of the above characterizations of knowledge topics applies to both target and project domains. As mentioned before, depending on their concerns, stakeholders may be interested in different knowledge topics.

## 6 Conversation Strategies

The knowledge transformations as discussed in section 4 are brought about by *conversations*. The scope of these conversations may range from ‘atomic’ actions involving a small number of actors, via discussions and workgroups, to the development process as a whole. Each conversation is presumed to have some *knowledge goal*: a knowledge state which the conversation aims to achieve (or maintain). This knowledge state can best be regarded as a multi-dimensional vector, positioning: (1) the knowledge topic; (2) the level of explicitness of the knowledge; (3) the level of sharing.

In achieving a knowledge goal, a conversation will follow a *conversation strategy*. Such a strategy is needed to achieve the goal of the conversation, starting out from the current state:

**Knowledge goal** – The knowledge goal; a desired knowledge state which the conversation will aim to achieve/contribute towards.

**Initial state** – The initial knowledge state as it holds at the start of the conversation.

Conversations take place in some situation in which resources may or may not be available for execution of the conversation. A conversation situation may be characterised further in terms of *situational factors* [FV99b]. We identify three classes of situational factors (each of which could be refined further):

**Availability of resources** – Refers to the availability of resources that can be used in a conversation.

**Complexity** – The resources needed for the conversation, the knowledge being conversed about, etc., will exhibit a certain level of complexity. This complexity also influences the conversation strategy to be followed.

**Uncertainty** – In determining a conversation strategy fit for a given situation, assumptions will have to be made about the knowledge goal, the initial state, the availability of resources, and the complexities of these factors.

In formulating a conversation strategy, all of the above factors should be taken into account. A conversation strategy should typically cover at least the following elements:

**Execution plan** – As mentioned before, a conversation can be composed of sub-conversations. Each of these sub-conversations focusses on a sub-goal, but they all contribute towards the goal of the conversation as a whole.

**Description languages** – The description languages to be used in the conversation(s).

**Media** – The kind of media to be used during the conversation(s).

**Cognitive mode** – The *cognitive mode* refers to the way in which knowledge is processed/gathered by the collective of actors involved in a conversation.

**Social mode** – The *social mode* is the way in which the actors executing the system development process collaborate with the actors from the business domain.

**Communication mode** – A small number of basic patterns of communication can be distinguished, as covered by combinations of the some basic factors: speaker-hearer ratio, requirements on hearer response, allowed time-lag, locality, and persistency.

## 7 Conclusion

As mentioned in the introduction, figure 1 depicts the resulting conceptual framework as it stands at present. Further work on refining, integrating, instantiating, and *validating* this framework is ongoing. We invite other scholars to join our efforts to better navigate the methodology jungle.

Currently, we are refining our results in two directions:

- a fundamental understanding of the use of modelling concepts for communicative purposes and
- refining our view on system development as a communication-driven knowledge transformation process.

Both research directions will involve fundamental theoretical activities as well as more practical oriented activities based on action research [Blu55, ALMN99, Bas99].

## References

- [ALMN99] D.E. Avison, F. Lau, M. Meyers, and P.A. Nielsen. Action Research. *Communications of the ACM*, 42(1):94–97, 1999.
- [Aus62] J.L. Austin. *How to Do Things with Words*. Clarendon Press, Oxford, United Kingdom, EU, 1962.

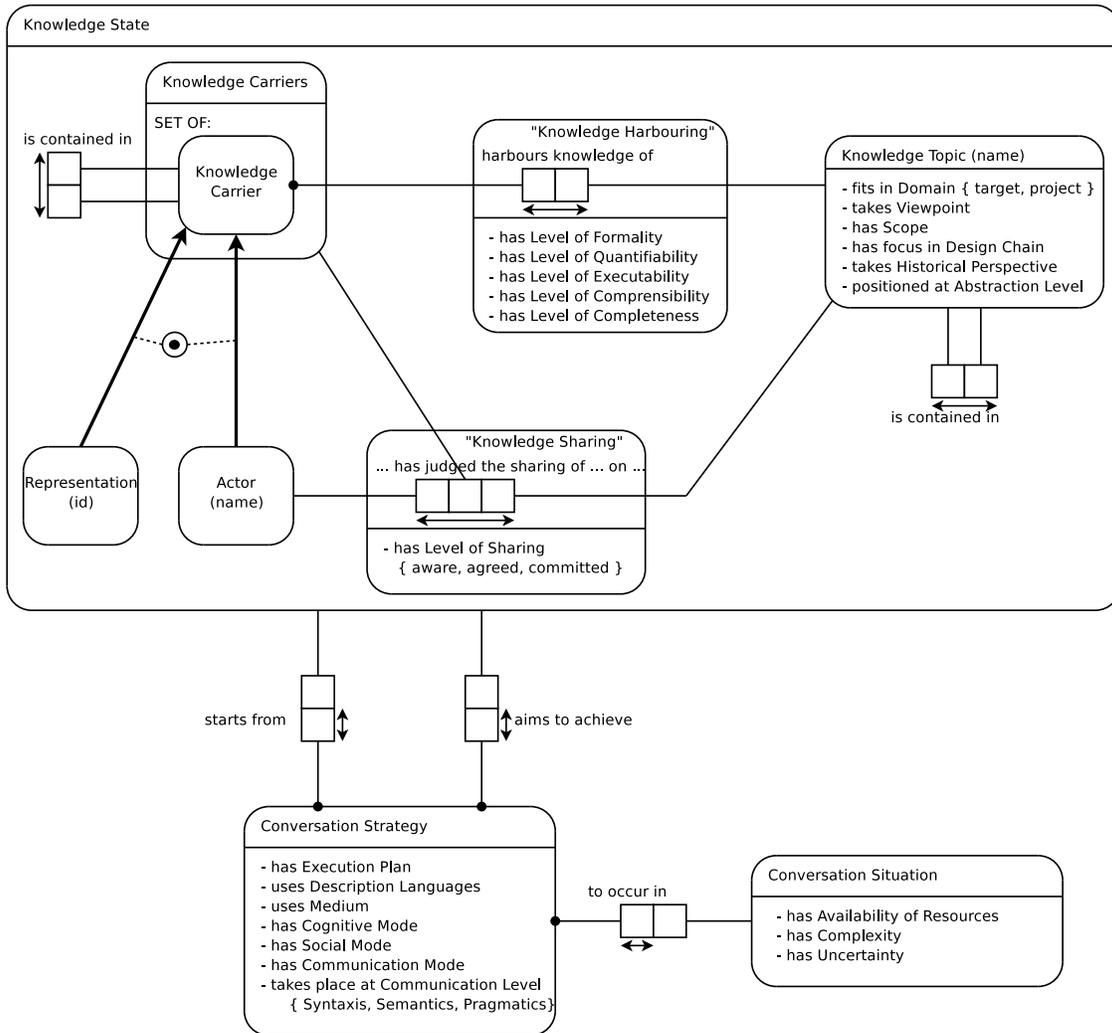


Figure 1: Resulting conceptual framework.

- [Avi95] D.E. Avison. *Information Systems Development: Methodologies, Techniques and Tools*. McGraw–Hill, New York, New York, USA, 2nd edition, 1995.
- [AW91] D.E. Avison and A.T. Wood-Harper. Information Systems Development Research: An Exploration of Ideas in Practice. *The Computer Journal*, 34(2):98–112, April 1991.
- [Bas99] R. Baskerville. Investigating Information Systems with Action Research. *Communications of the Association for Information Systems*, 2(19), October 1999.
- [BBMP95] G.H.W.M. Bronts, S.J. Brouwer, C.L.J. Martens, and H.A. (Erik) Proper. A Unifying Object Role Modelling Approach. *Information Systems*, 20(3):213–235, 1995.
- [BHW91] P. van Bommel, A.H.M. ter Hofstede, and Th.P. van der Weide. Semantics and verification of object–role models. *Information Systems*, 16(5):471–495, October 1991.
- [Blu55] F. Blum. Action research – A scientific approach? *Philosophy of Science*, 22(1):1–7, 1955.
- [BMS98] P. Bernus, K. Mertins, and G. Schmidt, editors. *Handbook on Architectures of Information Systems*. International Handbooks on Information Systems. Springer, Berlin, Germany, EU, 1998.
- [BPH04] A.I. Bleeker, H.A. (Erik) Proper, and S.J.B.A. Hoppenbrouwers. The Role of Concept Management in System Development – A practical and a theoretical perspective. In J. Grabis, A. Persson, and J. Stirna, editors, *Forum proceedings of the 16th Conference on Advanced Information Systems 2004 (CAiSE 2004), Riga, Latvia, EU*, pages 73–82, Riga, Latvia, EU, June 2004. Faculty of Computer Science and Information Technology.
- [Bri96] S. Brinkkemper. Method engineering: engineering of information systems development methods and tools. *Information and Software Technology*, 38(4):275–280, 1996.
- [Bub86] J.A. Bubenko. Information System Methodologies – A Research View. In T.W. Olle, H.G. Sol, and A.A. Verrijn-Stuart, editors, *Information Systems Design Methodologies: Improving the Practice, Amsterdam, The Netherlands, EU*, pages 289–318. North-Holland/IFIP WG8.1, Amsterdam, The Netherlands, EU, 1986.
- [CHP96] L.J. Campbell, T.A. Halpin, and H.A. (Erik) Proper. Conceptual Schemas with Abstractions – Making flat conceptual schemas more comprehensible. *Data & Knowledge Engineering*, 20(1):39–85, 1996.
- [CP96] P.N. Creasy and H.A. (Erik) Proper. A Generic Model for 3–Dimensional Conceptual Modelling. *Data & Knowledge Engineering*, 20(2):119–162, 1996.
- [Cru00] A. Cruse. *Meaning in Language, an Introduction to Semantics and Pragmatics*. Oxford University Press, Oxford, United Kingdom, EU, 2000.
- [DFW96] C.F. Derksen, P.J.M. Frederiks, and Th.P. van der Weide. Paraphrasing as a Technique to Support Object–Oriented Analysis. In R.P. van der Riet, J.F.M. Burg, and A.J. van der Vos, editors, *Proceedings of the Second Workshop on Applications of Natural Language to Databases (NLDB’96)*, pages 28–39, June 1996.
- [EKW92] D.W. Embley, B.D. Kurtz, and S.N. Woodfield. *Object–Oriented Systems Analysis – A model–driven approach*. Yourdon Press, New York, New York, USA, 1992.
- [FV99a] M. Franckson and T.F. Verhoef, editors. *Dictionary*. Information Services Procurement Library. ten Hagen & Stam, Den Haag, The Netherlands, EU, 1999.

- [FV99b] M. Franckson and T.F. Verhoef, editors. *Managing Risks and Planning Deliveries*. Information Services Procurement Library. ten Hagen & Stam, Den Haag, The Netherlands, EU, 1999.
- [FVV<sup>+</sup>98] E.D. Falkenberg, A.A. Verrijn-Stuart, K. Voss, W. Hesse, P. Lindgreen, B.E. Nilsson, J.L.H. Oei, C. Rolland, and R.K. and Stamper, editors. *A Framework of Information Systems Concepts*. IFIP WG 8.1 Task Group FRISCO, IFIP, Laxenburg, Austria, EU, 1998.
- [FW04] P.J.M. Frederiks and Th.P. van der Weide. Information Modeling: the process and the required competencies of its participants. In F. Meziane and E. M'etais, editors, *9th International Conference on Applications of Natural Language to Information Systems (NLDB 2004), Manchester, United Kingdom, EU*, volume 3136 of *Lecture Notes in Computer Science*, pages 123–134, Berlin, Germany, EU, 2004. Springer.
- [Hab84] J. Habermas. *The Theory for Communicative Action: Reason and Rationalization of Society*, volume 1. Boston Beacon Press, Boston, Massachusetts, USA, 1984.
- [Hal01] T.A. Halpin. *Information Modeling and Relational Databases, From Conceptual Analysis to Logical Design*. Morgan Kaufmann, San Mateo, California, USA, 2001.
- [HP98] A.H.M. ter Hofstede and H.A. (Erik) Proper. How to Formalize It? Formalization Principles for Information Systems Development Methods. *Information and Software Technology*, 40(10):519–540, October 1998.
- [HR00] B.-J. Hommes and V.E. Reijswoud. Assessing the Quality of Business Process Modelling Techniques: Introduction of the Q-Me framework and example. In *Proceedings of the 33rd Hawaii International Conference on Systems Sciences, Hawaii, USA*, Los Alamitos, California, USA, 2000. IEEE.
- [HVV97] J.J.A.C. Hoppenbrouwers, B. van der Vos, and S.J.B.A. Hoppenbrouwers. NL Structures and Conceptual Modelling: Grammalizing for KISS. *Data & Knowledge Engineering*, 23(1):79–92, 1997.
- [HW92] A.H.M. ter Hofstede and Th.P. van der Weide. Formalisation of techniques: chopping down the methodology jungle. *Information and Software Technology*, 34(1):57–65, January 1992.
- [HW93] A.H.M. ter Hofstede and Th.P. van der Weide. Expressiveness in conceptual data modelling. *Data & Knowledge Engineering*, 10(1):65–100, February 1993.
- [Kru00] P. Kruchten. *The Rational Unified Process: An Introduction*. Addison Wesley, Reading, Massachusetts, USA, 2nd edition, 2000.
- [Myl98] J. Mylopoulos. Techniques and Languages for the Description of Information Systems. In P. Bernus, K. Mertins, and G. Schmidt, editors, *Handbook on Architectures of Information Systems, Berlin, Germany, EU*. Springer, Berlin, Germany, EU, international handbooks on information systems edition, 1998.
- [NH89] G.M. Nijssen and T.A. Halpin. *Conceptual Schema and Relational Database Design: a fact oriented approach*. Prentice-Hall, Englewood Cliffs, New Jersey, USA, 1989.
- [OHM<sup>+</sup>88] T.W. Olle, J. Hagelstein, I.G. Macdonald, C. Rolland, H.G. Sol, F.J.M. van Assche, and A.A. Verrijn-Stuart. *Information Systems Methodologies: A Framework for Understanding*. Addison Wesley, Reading, Massachusetts, USA, 1988.

- [PH04] H.A. (Erik) Proper and S.J.B.A. Hoppenbrouwers. Concept Evolution in Information System Evolution. In J. Gravis, A. Persson, and J. Stirna, editors, *Forum proceedings of the 16th Conference on Advanced Information Systems 2004 (CAiSE 2004)*, Riga, Latvia, EU, Riga, Latvia, EU, pages 63–72, Riga, Latvia, EU, June 2004. Faculty of Computer Science and Information Technology.
- [Pro99] H.A. (Erik) Proper. Een derde dimensie voor informatiemodelleren. *DB/Magazine*, 10(3):52–55, May 1999. In Dutch.
- [PVH05] H.A. (Erik) Proper, A.A. Verrijn-Stuart, and S.J.B.A. Hoppenbrouwers. Towards Utility-based Selection of Architecture-Modelling Concepts. In S. Hartmann and M. Stumptner, editors, *Proceedings of the Second Asia-Pacific Conference on Conceptual Modelling (APCCM2005)*, Newcastle, New South Wales, Australia, volume 42 of *Conferences in Research and Practice in Information Technology Series*, pages 25–36, Sydney, New South Wales, Australia, January 2005. Australian Computer Society.
- [PW94] H.A. (Erik) Proper and Th.P. van der Weide. EVORM – A Conceptual Modelling Technique for Evolving Application Domains. *Data & Knowledge Engineering*, 12:313–359, 1994.
- [RB96] M. Rossi and S. Brinkkemper. Complexity Metrics for System Development Methods and Techniques. *Information Systems*, 21(2):209–227, 1996.
- [RMD99] V.E. van Reijswoud, J.B.F. Mulder, and J.L.G. Dietz. Communication Action Based Business Process and Information Modelling with DEMO. *The Information Systems Journal*, 9(2):117–138, 1999.
- [Sea69] J.R. Searle. *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press, Cambridge, United Kingdom, EU, 1969.
- [Sea79] J.R. Searle. A Taxonomy of Illocutionary Acts. In *Expression and Meaning: Studies in the Theory of Speech Acts*. Cambridge University Press, Cambridge, United Kingdom, EU, 1979.
- [Sea95] J.R. Searle. *The Construction of Social Reality*. Penguin Press, London, United Kingdom, EU, 1995.
- [VHP04] G.E. Veldhuijzen van Zanten, S.J.B.A. Hoppenbrouwers, and H.A. (Erik) Proper. System Development as a Rational Communicative Process. *Journal of Systemics, Cybernetics and Informatics*, 2(4), 2004.