INTERACTIVE PROCESS MODELS FOR KNOWLEDGE INTENSIVE PROJECT WORK

Håvard D. Jørgensen
Norwegian University of Science and Technology, and
SINTEF Telecom and Informatics.
http://www.informatics.sintef.no/~hdj/
hdj@sintef.no

Abstract. This extended abstract summarises the current state of a thesis project in information systems engineering. The overall problem is to support planning, coordination, management and performance of work in knowledge intensive projects. The approach is to apply *interactive process models*, created and updated by the project participants, to customise and control the behaviour of the computerised support system. A number of new and existing modelling techniques are combined to solve the identified problems of modelling by end users and flexible model-driven computer support. Model reuse is a core challenge both for increasing the usefulness of the models and for minimising the effort of modelling. An innovative framework that supports need driven model reuse has been designed. The contributions of this thesis are validated through prototype implementation, comparison with related work, and user experimentation in case studies. Concrete problem formulation and a definite target audience are identified as the main shortcomings at the current stage of the work.

1. Introduction

The problem of keeping conceptual models alive as active resources for work permeates information systems (IS) engineering. Enterprise models are created with great enthusiasm, but serve few purposes other than helping participants create a shared understanding and in some cases document a common terminology for the organisation. Workflow, ERP and software process support environments use models more actively to control work, but the rigidity of these systems have prevented their application to situated and evolving knowledge intensive projects. In software development, problems related to user involvement are more frequently met by prototypes and mock-ups than by conceptual modelling.

The aim of this thesis work is to identify approaches that can make conceptual models more useful for end users. The focus is on *interactive models*. An interactive model is made available to the end users by the information system in operation, and it partially controls the behaviour of the system. By altering the model, users can adapt the system to fit local needs, changes in the environment and new working practices. In particular, I study the use of interactive work process models to support project planning, management, coordination and performance in knowledge intensive industries.

2. Problem

While interactive models are becoming increasingly recognised as a promising technique for designing flexible information systems [5, 7, 20], challenges remain in transforming this general framework into practical design principles. This thesis explores the challenges facing the application of interactive models, and investigates which techniques can meet these challenges. In particular, this work contributes to understand and solve these problems:

1. *Articulation*
   What modelling language constructs are suitable for simple, user- and domain-oriented interactive models? What constructs makes it easy to adapt models that represent a changing environment?
2. *Activation*
   How can interactive activation (where users and IS components cooperate in situated interpretation of an evolving, incomplete model) be supported by an IS? How can incomplete models be enacted and how can different forms of model-driven functionality be combined without causing too complex models?
3. *Reuse*
   What are the suitable mechanisms for reusing interactive process models? How can local models be generalised and repackaged into templates? How can adaptation and composition of model fragments be supported as system services?

These questions are closely interdependent. User-oriented languages make it easier to learn how to articulate your work, while flexible modelling techniques enable model adaptation. This creates a potential for accurate models, which are needed for activation to function, and for reuse to serve knowledge management purposes. Activation makes the models useful, in that automatic customised support is offered to the users, while reuse simplifies the construction of new models, minimising the effort of articulation while increasing the usefulness of existing models.

2.1 General Approach: Interactive Models

In this section we elucidate the basic concepts and their relationships. Models are generally defined as explicit representations of some portions of reality as perceived by some actor [20]. A model is *active* if it directly influences the reality it reflects, i.e. if changes to the model also change reality or the way some actors perceive reality. Actors in this context include users as well as software components. *Model activation* is the process by which a model affects reality. Model activation involves actors interpreting the model and to some extent adjusting their behaviour accordingly. This process can be

- *Automated*, where a software component (e.g. a workflow engine) interprets the model,
- *Manual*, where the model guides the actions of human actors, or
- *Interactive*, where prescribed aspects of the model are automatically interpreted and ambiguous parts are left to the users to resolve.

Fully automated activation implies that the model must be formal and complete, while manual and interactive activation also can handle incomplete models.

Completing this conceptual framework, we define a model to be *interactive* if it is interactively activated. That a model is interactive entails a co-evolution of the model and the domain. A model that does not change will not be able to reflect aspects of reality that changes, nor can it reflect evolution of an actor's understanding. Consequently, an interactive model that does not evolve will deteriorate in representation quality. It contributes to change but does not reflect this change. Fully automated activation in a closed system can in theory avoid this deterioration. But cooperative IS are open systems, and interactive activation involves autonomous human actors, so the models will need to evolve. The process of updating an interactive model is called *articulation*. The interplay of articulation and activation reflects the mutual constitution of interactive models and the social reality they reflect (Figure 1). *Reuse* refers to applying a model (fragment) in a situation other than the one where it was originally constructed.

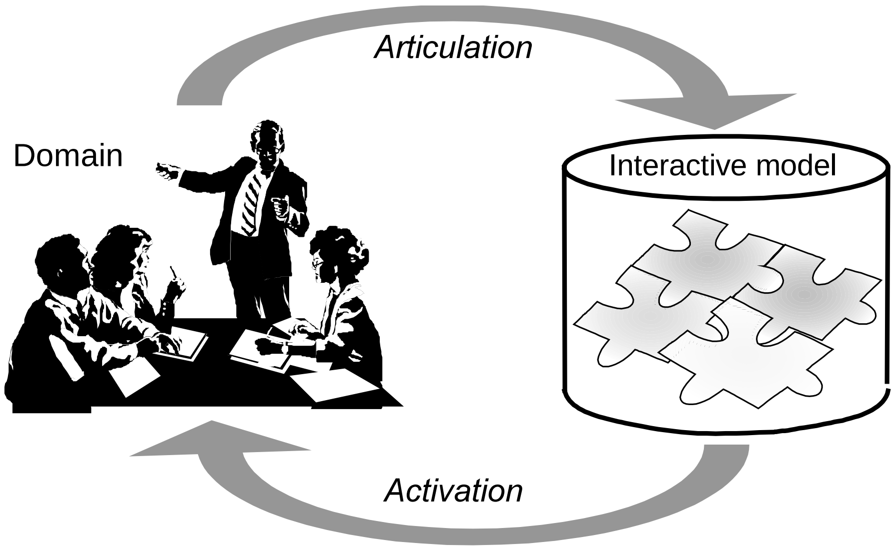

Figure 1. The interplay of articulation and activation.

3. Research Methodology

This thesis reports on an information systems *engineering* project. The approach is *constructive*, in that a prototype system is designed and implemented, and the major contributions are the design ideas and perspectives that were applied. The designs were developed in response to a wicked, open, incompletely understood, complex, real-world problem (supporting knowledge intensive project work). The approach is thus not reductionist in a scientific manner [4]. The objective was not to reduce the problem to a few precisely formulated and testable hypotheses. Engineering is about *design* of a solution to an open, holistic problem situation. Any attempt to formalise the problem entails an abstraction from the complexities of practice, removing some of the rationale behind design decisions. Problems are the sources of new ideas. By maintaining an ongoing *reflective dialogue* [18] with the problem throughout the project, I've had access to a richer set of idea triggers than what would be the case had the problem been formalised early on. Problem setting (framing, analysis, understanding) and problem solving are inherently intertwined in this reflective dialogue. Each new proposal can contribute to solving the current problems, but it also contributes to build an understanding of the problem, generating new perspectives and triggering new ideas for solutions. In software engineering, this intertwining of problem setting and problem solving is most evident in incremental and iterative development methods

In order to frame the problem (knowledge intensive project work), organisational and social science theories were studied, and requirements were collected from various settings. A prototype implementation serves as proof-of-concept for the design, while initial user experimentation illustrates its usability. The validation is described in greater detail in section 6.

4. Requirements

The term *knowledge intensive work* is a result of a classification of work by dominant resource, where knowledge intensive processes are distinguished from labour and capital intensive. Examples include consulting, engineering, software development and other forms of professional work [18]. Typically such work is non-routinised, dominated by information processing, dependent upon the knowledge and motivation of workers, demanding interdisciplinary and cross-functional cooperation, and causing empowerment and networked organisational structures rather than bureaucratic hierarchies. Knowledge intensive work is frequently organised in projects. A project is a temporary organisation established to handle a *unique problem*. Projects exist outside of functional organisation units, combining people from different units, often with complementary skills and knowledge that are needed for this particular problem. The situated, contingent, and emergent nature of project work requires flexible support infrastructures. Flexibility is seldom offered by the tools currently available for interorganisational cooperation, like e-business frameworks, workflow management systems, enterprise resource planning (ERP) etc. Contributing with design frameworks for more flexible, open information systems is thus a major objective of this work. The major requirements for utilising interactive process models to support knowledge intensive work, is summarised in the table below.

Requirement	
R1	Simple process modelling language (PML)
R2	Graphical PML, multiple views
R3	Domain-specific PML
R4	Concept evolution
R5	Local change, process history, versioning
R6	Dynamic incomplete models
R7	Personalisation and contextualisation
R8	Generalisation, analysis, metrics
R9	Structured template repository
R10	Description of suitable context of use, origin
R11	Extensible PML
R12	Multiple perspectives and interpretations
R13	Composition of template fragments

Table 1. Summary of requirements.

A survey of related work in workflow management and software process support [9] identified flexibility as a major limitation. Flexible model activation is primarily reflected in the requirements R3-R7, R12, and R13. Another key problem is that models do not reflect the real processes. This is partly due to limited flexibility, but also due to poor articulation support, hence modelling languages should meet requirements R1-3. The third major challenge is *reuse* (R8-R13). Current state-of-the-art in process model reuse is based on direct application of techniques from programming languages, e.g. inheritance and parameterisation, complemented with conceptual modelling techniques for e.g. composition. Typically these techniques are complicated and poorly understood by end users. Simplifications are needed if more user involvement (interaction) in reuse is to be facilitated. The unique and ambiguous characteristics of each project create increased demands for user involvement in model interpretation.

5. Related Work and Contributions

The related work for this thesis fall into four main categories:

- Interactive and active models
- Conceptual modelling
- Workflow management and process support systems
- Tailorable systems and end user computing

The thesis makes contributions to all of these areas.

5.1 Interactive and Active Models

Research into conceptual modelling has concentrated on IS *development* activities. Models of work environments have long been used to analyse the problem domain, to capture and structure user requirements. Chen et al. [5] points to active modelling as a major future direction for research in conceptual modelling, including model execution, end user participation, interaction etc. Greenwood et al. [7] argue that active models can enable IS to meet many business needs that current technologies fail to support. The primary focus of their work is a methodology for active software process support.

The interaction framework proposed by Wegner et al. [20], is the most comprehensive approach to this field. The development of this framework was triggered by the realisation that machines involving users in their problem solving can solve a larger class of problems than algorithmic systems computing in isolation. Hence, research should not solely be concerned with the development of more powerful algorithms, we should also look at new ways in which the computerised and human parts of the system can cooperate in order to solve problems. This thesis contributes with an elucidation of these theoretical concepts in a design perspective, pointing to interactive models as a general technique for designing flexible, user-friendly information systems. The project further uncovers requirements and challenges facing IS built on interactive models, and identifies suitable modelling techniques (which will be discussed below). To our knowledge, no such practical exploration of interactive models has previously been carried out. Thus, this thesis contributes with a much needed validation of the usefulness of the interaction approach for guiding IS design. The interaction framework is contested by some scientists because it does not really say anything new, theoretically that is. What it provides is a new perspective, a different way of framing, describing, analysing and solving problems [18]. Demonstrating its practical utility in design is thus fundamental.

5.2 Conceptual Modelling

To conceptual modelling in general this thesis contributes with an exploration of techniques suitable for interactive usage, especially targeting the problems concerned with end user articulation and model reuse. While some of the techniques listed below are not entirely new, their application to interactive models contributes to a more clear understanding of their benefits and limitations. In this context, the contribution of an IS engineering project will often be to identify suitable combinations of techniques, and to validate new and poorly tested techniques. Some techniques were also clearly extended by this work.

5.2.1 Holistic Semantics for Simple, Flexible Modelling Languages

Holistic semantics expresses meaning through combined interpretation of all the elements in a model. Popular modelling languages today provide limited degrees of holism, e.g. through inheritance, and rely mostly on atomic semantics, where the meaning of a model element is determined by that element alone. While the concept of holistic semantics was defined elsewhere [14], this work contributes with the identification of holistic modelling techniques like

- *Property modelling* [17], where properties can be dynamically attached to object instances,
- *Constellation modelling* (meaning is expressed by constellations of objects, each affecting the interpretation of the others), and
- *Context-sensitive semantics* (model elements change meaning according to the context where they are placed).

The process modelling language defined in this thesis demonstrates how these techniques can help us construct a simpler, more flexible modelling language, meeting requirements of concept evolution, multiple views, domain specificity, evolving models, extensible modelling language etc. Property modelling is especially useful to limit the complexity of the modelling language in cases where objects are classified along a number of independent dimensions. By defining one property for each dimension, instead of having separate class definitions for every combination of properties, the number of language concepts is greatly reduced. These features also enhance the flexibility of the models, as it is simpler to add or remove a property than to change the class of an object. Constellation modelling is similar to property modelling, only it is objects and not properties that are combined. Context sensitive semantics generalises constellation modelling in that other features than the objects grouped together can influence the semantics of an object. For instance, it can depend on the diagram type, e.g. objects appear differently in UML class diagrams and sequence diagrams.

Figure 2 below shows examples of these holistic modelling techniques,. The meaning of the person object differs from one contexts to another. A generic person object inside an organisation is interpreted as a *position*, when placed inside a task it denotes a *role* etc. We thus need not have separate constructs in the language for these concepts. The allocation of an individual person to a role (generic person object) is achieved by the addition of a *property* defining which person (e.g. name, email address, social security number). This corresponds to moving from the top line to the bottom in the figure. The addition of this property can be made directly or indirectly (through the establishment of a relationship called is-filled-by between a generic and a specific person). No matter which contexts the person objects appear in, there is thus a general way of moving from the abstract to the concrete. These state transitions are also achieved without altering the class of the object. In Figure 2 two or more persons placed together represent a group. This is an example of constellation modelling. The group can of course be placed in all the same contexts as single person objects, e.g. to denote group roles in a task, or teams in the organisation.

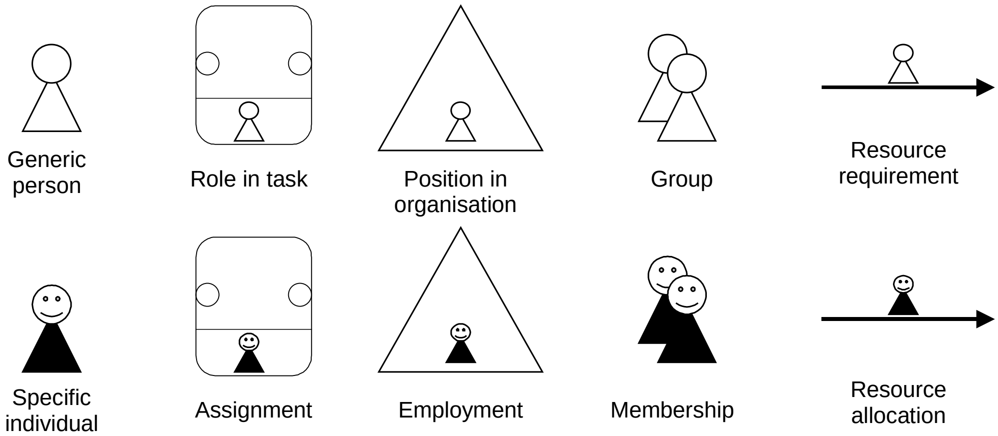

Figure 2. Holistic semantics of person objects.

All of these holistic modelling techniques contribute to simplifying the modelling language, by decreasing the number of concepts without compromising on expressiveness.

5.2.2 Holistic Activation Semantics

Another contribution of this thesis is the realisation that holistic semantics is a core part of a powerful interactive activation framework. An architecture where the interpretation of model elements depends on the *current states* of surrounding elements, enables a richer, more contextual and situated activation. For instance, the task in Figure 2 will be interpreted differently by the enactment engine depending on whether a person is already allocated to fill the role or not. The activation semantics emerges from the local interpretation of individual elements, better matching the contingencies of the work that the model reflects and supports [10]. In the thesis this general approach is exemplified by interactive workflow enactment, which will be discussed in section 5.3.

5.2.3 Instance-Based Reuse

Instance modelling has been proposed as a simpler and more flexible alternative to class modelling [17]. We use instance modelling to facilitate local change. A few instance-based programming systems and workflow applications exist, but comprehensive support for model reuse controlled by the end users, is not included in these systems. The main purpose of classification is to enable reuse. Most reuse strategies take a post-hoc documentation approach, asking "what have we learned from this project". The goal is to make components reusable at the time of their production, thus classification structures are static and an inherent part of component definition.

I argue that *need-driven reuse* (pre-hoc and ad-hoc), asking "what past experience is relevant for this situation", better matches the open, situated and emergent nature of cooperative knowledge work. A number of software services can aid the project participants in identifying models and model fragments from past projects that matches the current needs. There will of course be a number of template models especially adapted for reuse, but any past model that the users has access to is also reusable. These reuse mechanisms are based on instance inheritance. In this scheme, classification structures are dynamically constructed to meet local, evolving requirements. Classifications are separate from the core instance models. Classes can be defined by *intension*, e.g. a new subclass can be defined as all objects of a superclass that has a certain property, or by *extension*, where a superclass is defined by listing all its subclasses.

User-defined *reuse policies* control the propagation of reusable model aspects (properties, structure, behaviour) along modelled relations. Unlike most systems, where inheritance is coupled to specialisation relations only, my reuse framework supports reuse also along (de)composition relations, work flows, resource allocation and other interdependencies. The realisation that some model aspects should propagate along other relations than specialisation was triggered by specific user needs in our projects.

The notions of process model specialisation, completeness, and correctness are reinterpreted in this thesis [10]. An interactive model is complete only when all the work it represents has been performed, when it is no longer in use. This is the only time when it is no longer subject to the interplay of articulation and activation, when no more exceptions and further planning needs to be articulated. Semantic correctness of a model is no longer a prerequisite for its activation, as users, involved in interactive model interpretation, can handle ambiguities, incompleteness and inconsistencies manually when they perform the work that is incorrectly represented.

5.3 Workflow Management and Process Support Systems

The first contribution of this thesis project was the realisation that interaction is a suitable framework for flexible workflow systems [10, 11]. Viewing workflow enactment as an interactive process enabled us to provide much of the needed flexibility without increasing the complexity of the system and its modelling language. This complexity, caused e.g. by including detailed exception handling in the models and by requiring models to be semantically complete, made user involvement difficult.

- The difference between interactive and adaptive workflow [15] is described, clearing some of the conceptual confusion that has prevailed in this area, and pointing out directions for further research into interactive (also called human-centred, evolving and emergent) workflow systems [10]. While adaptive systems aim at making top-down workflow control more powerful, interactive workflow supports bottom-up articulation of work, enabling employee empowerment and partial group autonomy.
- *Interactive workflow enactment*: A scheme where users and IS components cooperate in interpreting the models in the situations that arise. The IS can

activate fully articulated model fragments, while human involvement is needed for ambiguous parts. Humans can also override prescribed behaviour when faced with unforeseen exceptions, through in situ articulation. Interactive enactment allows workflow instance models to evolve through different degrees of detail and specificity [11]. While a few existing workflow systems and research prototypes involve users in exception handling, we are not aware of any systems that apply interaction as the basic enactment mechanism and utilise this to simplify the modelling language, thus increasing the opportunity and likelihood of user involvement.

- *Explicit modelling of decisions* regarding the interpretation of the model. This technique enables interactive activation as decisions can be made either manually or automatically. It is shown how already existing model elements implicitly captures decisions, and that making these decisions explicit is a simple way of transforming a conventional modelling language into an interactive one.
- Policies that control model activation behaviour can be attached to model elements to tailor the system. A few existing systems apply policies to enable explicit reconfiguration of the enactment semantics. The holistic activation semantics described above is a type of *implicit*, partial and dynamic reconfiguration. The current state of the model determines which enactment rules are activated, so the users can tailor the system by altering the models of their work, in addition to explicitly selecting different policies.
- This thesis also extends the concept of enactment to include any kind of model-driven functionality, not just automated sequencing of work. Model driven document management, personnel allocation, communication support, access control and time management components have been proposed and to some extent implemented. The holistic semantics of the language and the proposed design enables the integration of several such model activators without unduly complicating the user interaction. The workflow architecture utilises constructs similar to aspect oriented programming [6] to achieve this.

5.4 Tailorable Systems and End User Computing

A wide variety of techniques have been proposed for building information systems, groupware and user interfaces that can be tailored by end users [13]. Interactive modelling allows user-oriented tailoring:

- *Domain and user-oriented language* concepts are used for tailoring the system. Most tailorable systems utilise a system-oriented language, allowing users to compose system components and combine system features. For example, with interactive models, the tailoring technique of *feature composition* is coupled to property modelling, i.e. when assigning a property to a model element the users also assign the behaviour that is attached to the property.
- *Integrating multiple model activators*. A model activator is a software component that (automatically or interactively) activates parts of a model. This thesis shows how multiple model activators, offering complementary interpretation of the same model, can be combined in an interactive architecture. This integration of multiple perspectives is enabled by holistic activation semantics, and demonstrated in the WORKWARE prototype (described below).

6. Validation

The contributions described above are validated in a number of ways. First and foremost, most of the ideas have been implemented in the WORKWARE prototype [11, 10]. This system combines interactive workflow enactment and with other forms of groupware functionality to provide richer and more flexible coordination support. It serves as proof of concept for interactive activation, the complementarity of multiple model activators and the underlying language formalisms. Anchored in the requirements presented above, comparisons with other systems demonstrate the innovative features of this work and its usefulness. In the EXTERNAL project[1], WORKWARE was integrated with number of other tools (model editors, visualisation, simulation, and communication) in a common infrastructure. The core of this infrastructure is a shared repository of interactive process models.

EXTERNAL applies this infrastructure in a number of case studies, representing different kinds of interorganisational knowledge intensive projects. So far we have concentrated on these processes.

- Quarterly progress reporting, a typical administrative, repetitive task,
- Joint project planning, slightly more creative and collaborative, and
- Software development subcontracting, which is even more complex.

Other members of the project are currently carrying out an evaluation of these experiences. Model examples collected so far illustrate the feasibility of the approach [10]. The reuse features will not be evaluated by the case studies, as they are not yet fully implemented. However, examples from the case studies will be used to illustrate the potential of the reuse approach.

The implementation and case studies will not sufficiently validate the generality of the approach. In order to do so, the language formalisms will be applied to redefine the generic enterprise modelling language of METIS (one of the EXTERNAL tools). This will show that the resulting language is simpler and more flexible. We will also investigate UML in this manner. In version 2.0 UML is moving towards more interactive features, increasing the focus on instances, inheritance of models and behaviour, increased compositionality etc. However, it currently seems lacking in simplicity and flexibility, precisely the main features provided by holistic semantics.

An extensive survey of organisation science literature [1, 2, 16, 18, 21, 22] also serves as a validation of the generality of the approach and solution. For instance, we uncover that socio-technical job design principles [19] are violated by conventional and adaptive workflow management systems, but not by interactive workflow. A wide variety of case studies of user participation in IS development, modelling, tailoring, process improvement, knowledge management and workflow evolution show that end user articulation is feasible, despite the scepticism of some researchers [3, 8]. It seems that this scepticism becomes self-fulfilling, in that systems are built, which do not facilitate articulation, hence no evidence of end user innovation can be captured by the system.

---

[1] EXTERNAL (Extended Enterprise Resources, Networks, and Learning) is an EU IST project investigating the use of interactive process models to support virtual enterprises. Its homepage can be found at http://research.dnv.com/external/.

7. Further Work

At the time of this writing, the thesis should be completed in 18 months. While a substantial part of the work has been completed, some challenges remain. Most notably, the problem must be made more concrete. While the current formulation gives direction and the wider context of the work, it is not operational in the sense that it can be verified that the thesis solves the problems. Some operational problem formulations are suggested below:

- *Articulation*
  Improve language efficiency (expressiveness divided by number of language concepts) and modelling flexibility (the number of operations that users must carry out in order to accomplish typical changes).
- *Activation*
  Increase the model driven functionality made available to the users without complicating the modelling language. Increase the support for end user involvement in situated model interpretation.
- *Reuse*
  Show that need driven process model reuse can be supported by an IS.

As discussed in 3, this is an engineering project. The problem formulation is a post-hoc rationalisation demanded by the standards of science. It is thus a major challenge to determine the suitable level of problem formalisation.

The identification of a primary target audience for this thesis is a related problem. In computer-supported cooperative work (CSCW), engineering research currently has a low standing [12]. Groupware and workflow management research is a clear candidate, but this field is currently too fragmented. Information systems is a wide discipline, and seemingly more concerned with systems development than with end user innovation during systems operation. Writing a thesis that will satisfy the detailed requirements of all these areas seems difficult.

References

1. Argyris, C. and Schön, D. *Organizational Learning: A Theory of Action Perspective*. Reading, MA, USA: Addison Wesley, 1978.
2. Berger, P. L. and Luckmann, T. *The Social Construction of Reality. A Treatise in the Sociology of Knowledge*. USA: Penguin Books, 1966.
3. Bernstein, A. *How Can Cooperative Work Tools Support Dynamic Group Processes? Bridging the Specificity Frontier*, Proceedings of CSCW 2000, Philadelphia, USA, 2000.
4. Bunge, M. *Philosophy of Science - Vol.1 From Problem to Theory*, Revised Edition ed. New Brunswick, USA: Transaction Publishers, 1998.
5. Chen, P. P., Thalheim, B. and Wong, L. Y. *Future Directions of Conceptual Modeling*, in Lecture Notes in Computer Science 1565, P. P. Chen, J. Akoka, H. Kangassalo, and B. Thalheim, Eds. Berlin, Germany: Springer, 1999.
6. Elrad, T., Filman, R. E. and Bader, A. *Aspect Oriented Programming - Special Issue*, Communications of the ACM, vol. 44, no. 10, 2001.
7. Greenwood, R. M., Robertson, I., Snowdon, R. A. and Warboys, B. C. *Active Models in Business*, 5th Conference on Business Information Technology, 1995.

8. Heinl, P., Horn, S., Jablonski, S., Neeb, J., Stein, K. and Teschke, M. A *Comprehensive Approach to Flexibility in Workflow Management Systems*, Proceedings of WACC '99, San Francisco, USA, 1999.
9. Jørgensen, H. D. *Software Process Model Reuse and Learning*, Process Support for Distributed Team-based Software Development (PDTSD'00) as part of SCI 2000, Orlando, Florida, 2000.
10. Jørgensen, H. D. *Interaction as a Framework for Flexible Workflow Modelling*, Proceedings of GROUP'01, Boulder, USA, 2001.
11. Jørgensen, H. D. and Carlsen, S. *Emergent Workflow: Integrated Planning and Performance of Process Instances*, Workflow Management '99, Münster, Germany, 1999.
12. Jørgensen, H. D., Krogstie, J. and Ohren, O. P. *The Future of CSCW: Scientific Maturity at the Expense of Interdisciplinarity?*, submitted to CSCW 2002, New Orleans, USA, 2002.
13. Kahler, H., Mørch, A., Stiemerling, O. and Wulf, V. *Special Issue on Tailorable Systems and Cooperative Work*, CSCW: The Journal of Collaborative Computing, vol. 9, no. 1, 2000.
14. Kangassalo, H. *Are Global Understanding, Communication and Information Management in Information Systems Possible?*, Lecture Notes in Computer Science 1565, P. P. Chen, J. Akoka, H. Kangassalo, and B. Thalheim, Eds. Berlin, Germany: Springer, 1999.
15. Klein, M., Dellarocas, C. and Bernstein, A. *Special Issue on Adaptive Workflow Systems*, CSCW: The Journal of Collaborative Computing, vol. 9, no. 3-4, 2000.
16. Orr, J. *Talking about Machines*. Ithaca, New York: Cornell University Press, ILR Press, 1996.
17. Parsons, J. and Wand, Y. *Emancipating Instances from the Tyranny of Classes in Information Modeling*, ACM Transactions on Database Systems, vol. 25, no. 2, 2000.
18. Schön, D. A. *The Reflective Practitioner*. Aldershot, UK: Ashgate Publishing Ltd., 1983.
19. Trist, E. L. *The Evolution of Socio-Technical Systems*, Issues in the Quality of Working Life, Ontario QWL Centre, no. 2, 1981.
20. Wegner, P. and Goldin, D. *Interaction as a Framework for Modeling*, Lecture Notes in Computer Science 1565, P. P. Chen, J. Akoka, H. Kangassalo, and B. Thalheim, Eds. Berlin, Germany: Springer, 1999.
21. Wenger, E. *Communities of Practice. Learning Meaning and Identity*. Cambridge, UK: Cambridge University Press, 1998.
22. Zuboff, S. *In the Age of the Smart Machine*. USA: Basic Books Inc, 1988.