Information Systems Frontiers 4:3, 257–272, 2002

Language/Action Meets Organisational Semiotics: Situating Conversations with Norms

Aldo de Moor
Infolab, Department of Information Systems and Management, Tilburg University, The Netherlands
E-mail: ademoor@kub.nl

Abstract. *Virtual professional communities require a strong co-evolution of their social and information systems. To ensure that the evolutionary process of their socio-technical systems is viable, a legitimate user-driven specification process is required. Such a process helps to ensure the meaningfulness and acceptability of specification changes. A specification method supporting this process should be grounded in the neo-humanist paradigm so that subjectivist and conflict aspects receive proper attention. Two related subfields of information science that have roots in this paradigm are the Language/Action Perspective (LAP) and organisational semiotics (OS). The RENISYS method for specification of the socio-technical systems of virtual professional communities is presented. It combines aspects from both LAP and OS, by building on work done in the DEMO (LAP) and MEASUR (OS) methodologies. It thus provides an operationalization of neo-humanist ideals that can help to extend theoretical and empirical research.*

Key Words. *virtual communities, socio-technical systems, neo-humanism, legitimacy, language/action perspective, norms, organisational semiotics, DEMO, MEASUR, RENISYS*

1. Introduction

Collaborative work is increasingly being done in a distributed fashion. People work together across sometimes great distances using readily available information technology like the Internet. However, in this collaboration participants do not act simply as individuals. Collaborators are dependent on one another as members of professional communities, in which they share goals, interests, and norms, among other things (Talbott, 1995).

We define the virtual professional communities in which this joint work takes place as communities of professionals whose collaboration on activities required to realize shared goals is mostly or completely computer-enabled. The technologies used often consist of commonly available information tools, such as mailers and web applications. In these communities, work is organised in the form of more or less structured workflows. A workflow can be defined as a recurring unit of work of which the coordination, control and execution can be partially or completely automated (de Moor and Jeusfeld, 2001). Many of the current workflow systems have an origin in logistics, which means that they create structures to implement and enforce frequently recurring processes. Ad hoc applications, on the other hand, focus more on supporting creative knowledge activities. Their main aim is to provide only partial control to ensure that tasks, responsibilities, etc. are delivered (Khoshafian and Buckiewicz, 1995). Such deliverables and control structures that act as boundary structures allow for more circumscribed process *spaces*, instead of rigid procedures (Fitzpatrick and Welsh, 1995). These spaces give workers certain degrees of freedom to situate their work according to their own needs and preferences. At the same time, they provide collaborators with some minimum level of guidance for their inter-dependent work. Typical examples of such hybrid workflow systems can be found in research networks publishing scientific publications, in which the rules for document production are very formal in terms of deliverables, sequences of activities, and responsibilities. On the other hand, the micro-structures and processes making up these workflows can be implemented in very different ways.

Ad hoc workflows being relatively underdefined has consequences for the design of the technical information systems supporting community members in their productive activities and interactions. Since this kind of work is so situated, requiring much additional human interpretation, traditional systems development methods grounded in the 'information flow' paradigm of symbol-manipulating functions are no longer sufficient (Stamper, 2000). Of course, we still need methods that describe in a precise way the social system, the supporting information system, and their linkages. Such analysis results in clear specifications of the configurations of the information tools that make up the technical implementation of the information system. At the same time, however, the methods should do justice to the subtle requirements and complex organisational behaviour of communities. Much of this complexity is caused by the important role that tacit knowledge plays in communities. This is the non-articulated knowledge that is embedded in the actions, experience and values of community members (Nonaka, Reinmoeller, and Senoo, 1998). In the development of these specification methods, we must therefore ask ourselves the important research question not only what kind of change the methods are to support, but especially *who* is to be involved in what way in these specification processes.

This article is organised as follows. In Section 2, we explain how the socio-technical system formed by professional community and its information system is prone to continuous change, and that a process of legitimate user-driven specification is necessary to support its evolution. Section 3 claims that methods supporting such a specification process should be grounded in the neo-humanist paradigm. Both LAP and OS are based on this paradigm, and have contributed to the development of the RENISYS method for legitimate user-driven specification of community information systems. In Section 4, we present a communications view on the specification process, showing how DEMO and its Transaction Process Model are adapted in RENISYS to model conversations for specification. Section 5 looks at how these conversations can be situated by a context of norms. To this purpose, the MEASUR method, an organisational semiotics representative, is analyzed. In Section 6, we illustrate the functionality of RENISYS by describing a use scenario, after which we end the article with some conclusions.

2. Changing the Socio-Technical System

From the previous, it follows that the development of information systems for virtual professional communities requires a continuous change process in which community members need to be actively involved. We now aim to establish more precisely the role of change in virtual professional communities. Change is continuous, should lead to a co-evolution of the social and technical system, and requires a form of legitimate user-driven specification.

2.1. Continuous change

Change is a fundamental concept in virtual professional communities. There are many change drivers in these complex socio-technical systems, including social, political, organisational and technical forces (Peterson, Smits, and Spanjers, 2000). Combined with the situatedness of work, these drivers result in a continuous pressure for adaptation of community information systems. Thus, waterfall-based approaches that result in complete and stable versions of large information systems are not very useful anymore (Brooks, 1995). Instead, methods that allow for the ongoing redefinition of specific parts of the community information system are needed.

2.2. Co-evolution of the social and technical systems

The social and technical domains are distinct, but related. The social (business) domain is the domain where requirements originate. In here, business processes and organisational structures are defined. The technical (IS) domain comprises the functionality that is provided by the various tools available to a community. After a lifetime of study, Engelbart sees these "human" and "tool" systems as together comprising an "augmentation system", which allows organisations to continuously improve their capabilities, provided that their co-evolution process is well managed (Engelbart, 1992). Different tools enable different information and communication processes: a word processor allows an individual to compose a document, a mailer enables somebody to send or receive an e-mail, and a mailing list permits a mail to be distributed from one sender to multiple receivers. In the design of the socio-technical system, requirements from the business domain are to be mapped to particular configurations of tool functionalities in the IS domain. Thus, a paper review workflow (business domain) could be enabled by a BSCW-server and web browsers (IS domain) in community A, while community B prefers to use mailers and a mailing list to organise its own version of the review process. In this way, complex dependencies can develop between requirements and tools. Whenever a change is proposed in either the requirements or technologies used, the

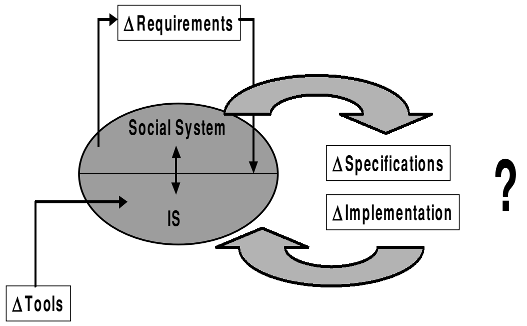

Fig. 1. *Co-evolution of the social and information system.*

method should therefore always allow for these dependencies to be traced, thus helping to ensure a true co-evolution of the two domains (Fig. 1). Some state-of-the-art workflow modelling methods, such as the ARIS toolset, exist that allow for such enterprise modelling, paying much attention to business components, technological tools and their interrelationships (Scheer, 1998).

Still, just knowing the dependencies between the two components of the socio-technical system is not enough. Over time, information technologies become an integral part of the meaning of the group as a dynamic social system (Hollingshead and McGrath, 1995). In methods like ARIS, the semantics of the changes are often specified in great detail. Nevertheless, it is often unclear who is authorised to make which specification changes, even though this may have quite detrimental effects on the evolution and performance of the system. Changes made under the control of an external design team are often opaque to the users, and may not reflect the interests of the community as a whole (de Moor and Jeusfeld, 2001). Thus, methods that concentrate strongly on the detailed *representation* of domain and process knowledge are no longer adequate. In a world in which socio-technical systems are characterized by large amounts of tacit knowledge, continuous change, and intricate community interactions the focus should shift from representing to *interpreting* this complex and often seemingly chaotic socio-technical reality.

The exploration of changes to such integrated and dynamic socio-technical systems therefore requires a subtle specification process, in which the meaningfulness and acceptability of changes are carefully explored. Thus, rephrasing the research question we asked ourselves in the introduction: how can we ensure that a community can only make changes to its socio-technical system that are acceptable within its field of social norms? To this purpose, specification methods should help in determining *who* should be involved in the continuous change process of the socio-technical system, preventing the production of specification changes that are artificial, obsolete, and alien to the community to which they apply.

2.3. Legitimate user-driven specification

Many specification methods (including traditional waterfall-based methods like ISAC, almost all workflow modelling methods, and even user-centered methods like ETHICS) assume that external analysts and designers, rather than the users, play the main initiating, coordinating or integrating roles in the specification process. From the previous, however, it becomes clear that for the kind of community information systems we are interested in, users, rather than just being consulted in the design process, should be true partners (Koh and Heng, 1996). Users first of all have the tacit knowledge that is often unavailable to outside observers. Second, they are the ones who face the breakdowns in work that lead to new insights about their true requirements or better designs of the tool configurations that are to enable these requirements (Winograd and Flores, 1986). Thus, the specification method should allow for *user-driven* development of the socio-technical system. However, these individual users are part of a community. This implies that changes to the socio-technical system proposed by one user, may affect the work of many other community members. Any proposed change must therefore be legitimate as well (de Moor and Jeusfeld, 2001). This entails first that a specification change must be *meaningful* in that its semantics are well-understood within the community. Second, changes need to be *acceptable* to the community. This means that the users to which the change is relevant, must agree, before it is implemented (Fig. 2).

3. Neo-Humanist Specification Methods

To construct new information systems development methodologies that support the legitimate user-driven specification process, we need to replace, or at least complement, the traditional waterfall and information flow paradigms. Neo-humanism provides us with this new worldview (Hirschheim, Klein, and Lyytinen,

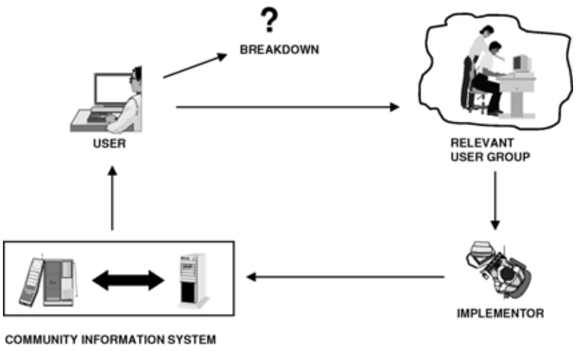

Fig. 2. *The legitimate user-driven specification process.*

1995). This paradigm is characterized by subjectivism and conflict. The subjectivist perspective holds that knowledge is socially constructed in a process of human interaction. The conflict view assumes there is a natural tendency towards change and conflict. This paradigm is especially suited to model information systems development in virtual professional communities, because in those communities stakeholders with many conflicting interests need to work together and construct joint models of their work processes and supporting information technologies.

The core idea of neo-humanism is its focus on *emancipation*, which is the process in which pseudo-natural constraints on the realization of human needs and potentials are removed by conscious attempts of human reason (Hirschheim, Klein, and Lyytinen, 1995). *Pseudo-natural constraints* are those that seem natural, but in fact are caused by communication distortions. Some of the most common reasons for such distortions to occur are authority and illegitimate power; peer opinion pressure; time, space, and resource limitations; social differentiation between actors; and bias and limitations of language use (Hirschheim and Klein, 1994). All of these distortions can seriously jeopardize the success of a virtual professional community, and it is therefore essential that they are being dealt with.

In neo-humanist system development approaches, the removal of such communication distortions is to be achieved by enabling what is termed *rational discourse*. In such discourse, claims made throughout the system development process are critically evaluated (Hirschheim and Klein, 1994). A process of rational discourse is thus essential if (sustained) collaboration and specification in virtual professional communities is to be achieved.

Although the ideas underlying neo-humanist development approaches are sound, they are not without limitations. One major drawback is that there are currently very few examples of how the neo-humanist ideals are to be put in practice (Hirschheim and Klein, 1994). For instance, it is not clear how the system development life cycle should be modified to promote emancipatory discourse, and how participants can be motivated to take part in such debate (Hirschheim, Klein, and Lyytinen, 1995).

3.1. LAP and OS

Two streams of research grounded in the neo-humanist paradigm are the Language/Action Perspective (LAP) and organisational semiotics (OS). Both fields stress the organisational instead of the IT aspects of information systems. Furthermore, language plays a crucial role in the execution and coordination of organisational work. LAP considers language not only as a means to exchange information, but also as a way to perform actions. It therefore studies information systems from the perspective of the conversations that are being conducted to get things done. OS is the study of organisational signs and signifying systems, analyzing a wide range of semiotic aspects of organisation and technology. Although they have different ways of looking at the same organisational information systems, both fields can and should be linked. LAP provides a focused analytical perspective through its conversational model, while OS integrates ideas from many different fields through its sign constructs and levels. For example, in order to produce correct illocutions in LAP models, the knowledge about relevant roles, relationships, resources, and histories that is provided by OS-theories such as the semiotic ladder (which distinguishes between semiotic levels ranging from the physical to the social layer) is indispensable (Stamper, 2001).

Not many attempts have yet been made to systematically combine the strengths of LAP and OS-approaches. One notable exception is Van Reijswoud's Transaction Process Model, which provides an analytical instrument for business communication by combining work on discourse communication models with discourse norms (van Reijswoud, 1996). One domain still uncovered is that of methods for the legitimate user-driven specification of community information systems. The purpose of the RENISYS (REsearch Network Information SYstem Specification) project was to develop such a method (de Moor, 1999; de Moor and Jeusfeld, 2001).

3.2. The RENISYS method: Situating conversations for specification

The RENISYS method allows individual users who have become aware of a problem with either the way their work is organised, or with the support provided by the enabling information technologies, to formulate their problems in terms of problematic *knowledge definitions*. A knowledge definition describes part of the structure or behaviour of the socio-technical system, for example a workflow, an information tool used, or a norm regulating the behaviour of community members. Four types of knowledge definitions are distinguished in RENISYS: type definitions ("a mailing list is an information tool that ... "), state definitions ("John is the editor of the ISF journal"), action norms ("An editor is permitted to execute the reviewer assignment process"), and composition norms ("An editor may initiate the modification of editorial process definitions"). More will be said about the roles of these definitions in the specification process in Section 5.

Once a breakdown has been formulated by the user experiencing it, RENISYS determines which other users are to be involved in the resolution of these definitions. To this purpose, the *composition norms* that regulate the acceptable specification behaviour of actors (or stakeholders) in the community play an important role. An example of such a norm would be that the editorial board is permitted to create new editorial workflows. The method calculates the *resultant deontic effect* of the set of composition norms that apply to the combination of a particular user and the specification process required to change the definition (e.g. for John/execution of workflow type modification). In this way, it knows which users to involve in the *conversation for specification* (i.e. the workflow definition discussion) in which the *problematic knowledge definition* can be legitimately changed. Additionally, or alternatively, a *discourse process* may be started in which users can critically examine background assumptions that determine the meaning of the various knowledge definitions making up the system specifications.

The functionality of the specification method and prototype tool were already extensively discussed in de Moor (1999) and de Moor and Jeusfeld (2001). In this paper, we investigate the way the method is grounded in both LAP and OS theory. In this way, we intend to strengthen the case for more research that (re)combines and applies these theoretical concepts to the development of new humanistic information systems methodologies. This in turn may help to show the usefulness and power of neo-humanist work to the mainstream of IS research, contributing to a much needed dialogue with the more established branches of information science.

4. A Communications View on the Specification Process

In this section, we briefly outline the LAP origins of RENISYS. Section 4.1 describes how the specification process can be regarded as a conversation between community members. Section 4.2 outlines Van Reijswoud's Transaction Process Model. Section 4.3 explains how we adapted that model to structure our conversations for specification.

4.1. Conversations for specification

In LAP, activities are generally analyzed using work based on Searle's speech act theory (Searle, 1969). However, the use of individual speech acts is insufficient to coordinate meaningful work-related communication. To do so, larger units of communicative interaction are needed, which are called *conversations*. In this paper, we adopt a somewhat restricted view on conversations, seeing them as a series of interrelated communicative acts aimed at defining and reaching a goal (Dietz, 1994). Taking into account the purpose of this paper, we define a *conversation* as a self-contained unit of communication to accomplish certain specification objectives, like the specification of a new workflow type. Evidence for the effectiveness of predefined conversation models is ambiguous (Suchman, 1994; Auramäki and Lyytinen, 1996). We therefore require a conversation to be only partially structured in the sense that main specification process entities are predetermined, although the format of the utterance acts in which these entities are defined is relatively free. There are many types of work-related conversations, one of which is the conversation for action, in which the goal is to coordinate explicit cooperative action (Winograd, 1987). This kind of conversation is the basis for the well-known Coordinator and ActionWorkflow modelling methods (Medina-Mora et al., 1993).

Many different types of conversations, but especially the conversation for action, can play some role in the specification process. This process is triggered by breakdowns in work, in the sense of Winograd and Flores (1986). As a consequence of the occurrence or anticipation of breakdowns, new semantic distinctions are always emerging. The generation and interpretation of these distinctions should be treated as an activity based on conversations that can be designed and facilited through the computer (Winograd, 1987). To do so, a conversation framework is needed that combines specialized as well as more general conversation patterns that can provide support for unanticipated breakdown-initiated conversations (Kensing and Winograd, 1991). We call such a conversation, which may be constructed out of a number of the abovementioned more or less structured conversations, a *conversation for specification*.

4.2. The transaction process model

Two key conversational roles are distinguised in LAP: the *speaker* and the *hearer*. It is often a problem for the hearer to classify the illocutionary force (i.e. the rationale) of an utterance made by the speaker. It must thus be made clear in any conversational state which (finite) set of *conversational actions* or *moves* are possible (Winograd and Flores, 1986; Schäl, 1996). In addition to the modelling techniques such as used in the conversation for action approach mentioned before, theory-grounded *conversation protocols* are therefore needed that can prescribe the allowed conversational moves for the participant whose turn it is to speak. One such protocol, aimed at modelling the mutual agreement dimension of conversations, is Van Reijswoud's Transaction Process Model (TPM) (van Reijswoud, 1996).

The TPM, itself based on Dietz's Dynamic Essential Modeling of Organisations (DEMO) method (Dietz, 1994), is a communication model that presents the possible conversational moves in a business communication process, thus providing a full understanding of the activity coordinating nature of DEMO's main idea, the transaction concept. The model is represented as a state transition diagram in which the states represent *transaction states* and the transitions are caused by *transaction acts*. These acts are subdivided into two categories: communication acts and objective acts. A *communication act* is an utterance by a participant that causes a transaction process transition. An *objective act*, the purpose of the transaction, is the act that changes the objective world. Objective acts do not need to be further modelled, as the actual activities that change the objective world are not part of the communication process.

Whereas the state transition technique is generally applied to modelling the behaviour of objects in the object world (the world of "things"), in the TPM it is used to represent the communication behaviour of subjects in the intersubject world (the world of "communicating people"). Besides being able to model successful communication processes, the TPM also allows for the representation of discussion and discourse, as proposed in Habermas's theory of communicative action (see White (1988) for a good summary of this theory). The TPM therefore consists of three layers. In the *success-layer*, a regular transaction process is described. The *discussion and failure-layer* allows for the discussion of validity claims, such as to the sincerity of the speaker. The *discourse-layer* models discourse with the purpose of restoring background conditions, by allowing for the questioning of assumptions. The discussion-layer can only be entered after communication in the success-layer has taken place, whereas the discourse-layer cannot be invoked before communication has occurred in the other two layers.

4.3. RENISYS: The specification process model

Although the TPM forms the basis for the conversation protocols needed in RENISYS, there are certain differences in terminology and application. We therefore use the term *Specification Process Model* (SPM) for the conversation model used in RENISYS. The main differences with the TPM are that the transaction is renamed into *specification process* and that the *evaluator* role is added to the existing initiator and executor roles. The purpose of the specification process is no longer an 'objective action', but a *definition process*. Communication acts and transaction states are renamed into the more precise terms *conversation acts* and *conversation states*. Other differences, notably the different interpretation of validity claims are discussed in de Moor (1999).

In the SPM, we want to formalize conversations as little as possible, in order to provide flexibility and not to cognitively overburden users. In this way, we heed the justified warnings of imposed conversational structures easily becoming too rigid and constraining on human communication (Suchman, 1994). Thus, although a user can start a discussion to, say, question the sincerity of another user's conversation act, the initiator does not need to formally indicate why he does so. The reason for this is that RENISYS *enforces* the legitimacy of conversations for specification by only inviting participants who are acceptable, as determined by the roles they play and the communal norms that apply. Once selected, participants are free to discuss in any way they like. This is a new application of the TPM: whereas

Table 1. *The conversation acts and definition process in a successful type creation process*

Act	Description	Resulting state
CA_1:	I: C_{legit}[propose(directive)<created_type_def(edit),now>]	Directed
CA_2:	X: C_{legit}[promise(commissive)<created_type_def(edit),now>]	Committed
DP:	X: DP_{legit}[define(execute)<created_type_def(edit),now>]	Executed
CA_3:	X: C_{legit}[report_completion(declarative)<created_type_def(edit),now>]	Declared(Completion)
CA_4:	E: C_{legit}[declare_success(declarative)<created_type_def(edit),now>]	Declared(Success)

the latter is a model to *analyze* conversational moves, RENISYS uses its SPM to *select* participants who are to take part in them. Thus, we use the SPM in a prescriptive, rather than a descriptive way.

To illustrate the use of the SPM, the conversation acts of a *successful specification process* to create an 'edit' workflow (type) definition are presented here, similar to the examples given in van Reijswoud (1996, p. 95). A complete overview of the conversation acts making up the SPM is given in de Moor (1999). The sequence of conversation acts (and the definition process they embed) is shown in Table 1.

The meaning of the table is that the initiator can legitimately propose to the executor to define an editorial (type) definition (CA1). The executor can then promise to do so, creating a commitment (CA2), etc.

5. Situating Conversations for Specification with Norms

One major criticism of the application of speech act theory to systems development is that it is not able to represent what people really do, as it provides models that are too rigid and simplistic to capture the complexities of actual work practices (Suchman, 1994; Auramäki and Lyytinen, 1996; Verharen, 1997).

Thus, in real social practice, the complex world beyond the representations must somehow be considered. In other words, it is not just important to produce definitions, but also to understand the *situatedness* of the conversations in which the definitions are produced, the way in which the definitions are represented and how they are understood by the people who use them (Winograd, 1987; De Michelis and Grasso, 1994; Taylor, 1998). Thus, a fundamental problem has not been addressed by the TPM (and, so far, by the SPM): how to make the link between the specific 'social/organisational and work situations' and the conversations for specification? Refining the research question about who to involve in the change process of the socio-technical system that we asked ourselves in the beginning of this paper:

> *Who are to be the initiators, executors, and evaluators of these conversations and what should be on their agendas?*

To this purpose, it is important that the *context* of the conversation for specification is taken into account (De Michelis and Grasso, 1994). However, the idea of context in speech act theory is still only vaguely defined and it is not yet very clear how it is to be used in systems development (Verharen, 1997). A major implication of LAP is that context interpretation cannot be fully automated, but to a large extent remains to be done by persons (Hanseth, 1991; Weigand and Dignum, 1997). Selecting the right community members to take part in the specification process is thus key.

Section 5.1 explains the conversation context model used in RENISYS. Sections 5.2 and 5.3 discuss the MEASUR approach to normative IS development and how it was used to develop a normative framework for RENISYS.

5.1. RENISYS: The conversation context model

Having defined the LAP-elements of RENISYS, we can define the meaning of context in our approach (Fig. 3). It consists of two parts, the internal and external conversation context.

Internal conversation context: the knowledge definitions which are semantically related to the knowledge definition being changed.

External conversation context: the knowledge definitions needed to select the users who can legitimately be involved in a particular conversation for specification.

Our internal conversation context gives meaning to the knowledge definition being changed, by situating it in a web of semantically related definitions, that are already meaningful and acceptable to the community.

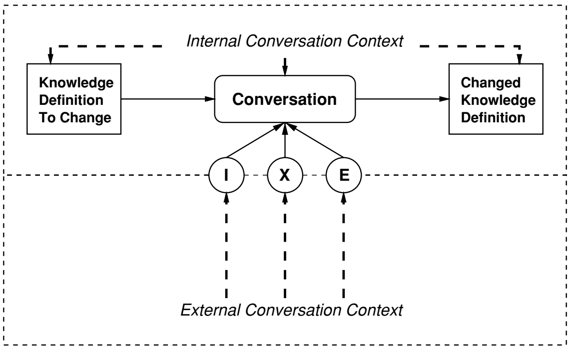

Fig. 3. *The context of conversations for specification.*

The external conversation context is similar to Taylor's 'institutional context'. He sees each speech act as being part of an indefinite series of interactions. The sum of past speech acts creates an institutional grounding for currently acceptable actions. Current speech act-based methods, such as DEMO, however, largely ignore this role of the institutional context in an ongoing work conversation (Taylor, 1998).

In order to model internal and external conversation contexts, different kinds of *knowledge definition categories* are necessary. Earlier in this paper, we introduced the four kinds of knowledge definitions that are distinguished in RENISYS (de Moor and Jeusfeld, 2001): *type definitions* determine the ontological meaning of concepts, while *state definitions* capture states-of-affairs. Furthermore, there are two kinds of norms, action norms and composition norms. *Action norms* specify acceptable operational behaviour, e.g. an author is permitted to submit a paper. *Composition norms*, on the other hand, specify acceptable change behaviour. These meta-level norms are essential for ensuring the legitimacy of changes to the socio-technical system. To discuss the properties of these norms, we turn to one of the most comprehensive methodological approaches for norm-guided systems development in organisational semiotics: the MEASUR methodology.

5.2. MEASUR: A norm-based specification approach

Ronald Stamper has initiated the MEASUR research programme, based on his semiotic theory, in which a range of methods and techniques for requirements analysis, systems design and systems construction has been developed (Stamper, 1992, 1994, 1996; Liu, 1993). The field of *semiotics*, itself founded by Charles Peirce at the end of the last century, provides the idea of the sign as a primitive notion upon which more complex concepts like information and communication can be built. If this idea is applied to information systems development, then modelling an information system can be regarded as representing an organisation in which people use signs for business purposes. Such a socio-technical perspective thus considers the organisation as a whole to be the information system.[1]

MEASUR adopts a radical subjectivist paradigm, in which there is (1) no knowledge without a knower and (2) no knowing without actions (by participants, observers, or receivers of reports). The most fundamental concepts of the theory are affordances and norms. An *affordance* is a universal invariant which constitutes the repertoire of an agent's behaviour. For example, a user (agent) plus a web browser (environment) afford surfing the Web. A *norm* is a social affordance, an affordance which has been accepted by a community as common ground. Norms provide the socially acceptable boundaries of behaviour.

A key assumption underlying MEASUR is that an information system is a socially constructed system that to a large extent is informal. Rather than focusing on abstract process specifications, it concentrates on the specification roles of individual agents, acting in the 'information field' produced by the shared norms of a social group. The interplay between these norms results in a certain resultant 'force' determining the acceptable behaviour of particular actors in a specific community (Stamper, 2000).

The requirements analysis part of the MEASUR programme is formed by the Semantic Analysis and Norm Analysis activities. Semantic Analysis first defines a problem in natural language, after which candidate agents and affordances are identified and grouped. These steps are followed by the ontological charting, in which all the groupings are stored in a general semantic model. This permits the definition of powerful structural constraints on allowable ontological relationships. Such *ontological constraints* define the basic 'possible' world. The next stage, Norm Analysis, then describes the deontic world, which comprises the socially-determined repertoires of behaviour that agents may adopt. The specification language used in these analyses is NORMA (Norms and Affordances), which contains a large number of affordance types. The norms are defined as constraints on the realisations of the affordances, stating under what conditions such ontological elements can start and stop to exist.

Such norms specify the authorities, either responsible agents or other norms, that govern the start and finish of such elements.

The ontology charts and norms are stored in the Semantic Temporal Database. This database contains affordances, determiners, and particulars. *Determiners* are invariants of quality and quantity that differentiate one instance, a *particular*, from another. The database can manage the temporal dynamics of the stored information using the database language LEGOL. Using this language, one can formally specify norms so that they can be used as database constraints, or to trigger actions.

5.2.1. Ontologies and norms in MEASUR. The *ontologies* that Semantic Analysis produces, assume that the world that the particular agent knows of comprises only those actions he can perform in his environment. An *ontological dependency* occurs when one affordance is only possible while another affordance exists. *Agents* themselves are also affordances, but of a special kind, namely those affordances which are able to act responsibly.

An example of an affordance would be:

```
(author draft) revisedraft
```

which means that revising a draft is ontologically dependent upon there being an author as well as a draft.

Norms provide guidance for actions. They can be seen as collective affordances of the complex agent at the social level, thus representing behavioural options that are socially determined (Liu, 1993; Stamper, 1994). In MEASUR, norms have a wide range of meanings. In their most generic interpretation they have the following format:

```
<condition> → <consequent>
```

Here, the condition is some perception of the situation, and the consequent some effect that is to occur, for instance, that an action should be taken by an actor.

Two examples of such norms are the following (Liu, 1993, p. 54):

```
1. author(paper#selected) → eligibi-
   lity#priority#1
2. 6 months before start-of meeting
   #CRIS-2 while (selected(paper)
   while-not invited(author(paper)))
   → print author
```

The first norm says that an author of a selected paper is assigned with first priority. The second norm is a 'trigger norm for actions', which says that half a year before the conference the name of any author of a selected paper who has not been invited needs to be printed.

A more detailed norm schema is the following (Stamper, 1992):

if <condition> **then** <some agent> **is permitted/forbidden/obliged to do** <action>

In this schema it is clearer that the norm attaches some deontic status to an agent-action pair. It is this definition of norms that is most relevant to the purposes of RENISYS.

5.2.2. Norm classifications. In MEASUR, various norm classifications are used.

One distinction is between perceptual, cognitive, evaluative, and behavioural norms. *Perceptual norms* define ways in which people can see the world, allowing them to meaningfully communicate about situations and events in the world. *Cognitive norms* are standardised beliefs and knowledge possessed by a group that foster knowledge sharing. *Evaluative norms* provide criteria for the assessment of people's behaviour. *Behavioural norms* govern people to behave in an appropriate manner in a given cultural setting.

A second classification of norms is on the formality dimension. *Informal norms* do not have a formal representation, whereas *formal norms* have been officially documented.

Thirdly, a distinction is made between explicit and implicit norms. An *explicit norm* is a norm that has been communicated and agreed upon, possibly vocally, whereas an *implicit norm* is a convention which has never been discussed, but to which the members of a community adhere.

5.2.3. An analysis of MEASUR. A major strength of MEASUR is its capability to define complex domain ontologies and norm bases. It facilitates the resolution of ambiguities in concepts that arise during their use by putting them into a context of concepts that are understood by the users. Another important feature is that informal norms are recognized to play an important role. Rather than attempting to make them all explicit, which is a practical and philosophical impossibility, the specification method attempts to link to informal norms by identifying the particular human agents responsible for their application as members of the relevant social group (Stamper, 1992). Such agents can then scan the information field for relevant information outside the scope of the computer system. Especially in CSCW systems, as compared to administrative systems, there is a high proportion of such implicit yet essential norms (Stamper, 1994).

On the other hand, the method also has its drawbacks with respect to the facilitation of the legitimate user-driven specification process. One of the basic assumptions made by RENISYS, shared by MEASUR, is that users themselves define their own problems, in their own terminology. However, a major drawback of Semantic Analysis is that it does not provide explicit guidance regarding what is to be modelled in case of a breakdown. This method in practice still requires the involvement of an external analyst who controls the modelling process, as it assumes many complex analytical skills to be present. In RENISYS these skills are to a large extent provided by the method (1) selecting the group of users relevant to a particular specification problem and (2) presenting them with the most relevant context knowledge.

Another issue is that MEASUR does not make an explicit distinction between norms that regulate operational and specification behaviour, which RENISYS does. This makes it difficult to express specification problems. This limitation is exacerbated when norms of different levels of generality and categories apply simultaneously, as their completeness and consistency cannot be guaranteed in MEASUR. Even more complexity is introduced if the *meta-norms* that guide the specification process themselves are also subject to change, as is the case in real-work network evolution. Furthermore, explicit procedures for users to discuss and resolve specification conflicts are not available.

Summarizing, from MEASUR the following elements are adopted in RENISYS:

- The distinction between ontological and normative knowledge. Ontologies can be used to define the properties of concepts in both the real-world and the information system that represents them (Wand et al., 1995). Ontological definitions of specification process entities are required before norms related to these entities can be defined. They thus form constraints on the possible norms. To illustrate, by asking "*what* is the editorial process?", one is interested in ontological properties of this process, by asking "*who* may execute the editorial process?" the focus shifts to the normative aspects of this process entity.
- The *deontic effect* classification of norms. According to this classification, norms are either *permissions*, *obligations*, or *prohibitions*. Since we are mainly interested in collaboration in professional communities, obligations are directed to particular persons and will therefore be referred to as *responsibilities*.
- The recognition of informal norms. The specification of many norms does not have to be worked out in every detail. Often, it suffices, or it is even the only feasible thing to do, to identify which actors are to interpret a particular work situation, instead of exhaustively defining the norms.

The ontological and normative approach developed in RENISYS differs from MEASUR in several important respects as well. RENISYS does not support the complete traditional information system development process that MEASUR does. Instead, we adopt, simplify, and extend some of its core ideas, notably on ontologies and norms, that are useful for our particular purpose of legitimate user-driven specification. RENISYS contains:

- An extra classification of norms according to the process level of work-related communication. On the one hand there are the norms that guide the operational work processes of network participants, called *action norms*. The second category of norms, called *composition norms*, are norms that guide the conversations in which network participants produce the specifications of their information system and its context.
- A different subdivision of what is ontological and normative knowledge. In MEASUR, ontologies only contain physical dependencies between entities, whereas norms regulate the conditions under which such entities can come into being and removed. So-called 'detailed norms' can also trigger actions fully automatically, which is necessary for the representation of implementation details. In RENISYS, the ontologies contain definitions of the properties of actor, object, and process entities, including of the events that automatically trigger actions in workflows. RENISYS ontologies thus describe the entities that play a role in its norms, and also comprise, for instance, the MEASUR perception norms. The RENISYS norms, however, are more specific, in the sense that they only focus on the distinction

between work and specification processes, whereas MEASUR offers a wide range of norm categories that are less relevant for this purpose. For instance, a RENISYS action norm can express that some actor is permitted to evaluate the results of a particular workflow process. Additionally, a MEASUR evaluative norm could be used to indicate how exactly this evaluation process is to be performed. Such distinctions, however, are too detailed for the purposes of RENISYS, as we are mainly interested in the specification of responsibilities, and not of detailed operational procedures.

- Different formal representations of ontological and normative knowledge. The MEASUR knowledge representation format is optimized for its Semantic Temporal Database. The RENISYS knowledge format is tailored to regulating specification conversations. Its format is simpler, as norms always apply in the here-and-now, whereas MEASUR allows for complex temporal constraints to be represented.
- RENISYS represents ontological concepts in a type hierarchy. This allows for generalizations and specializations of specification knowledge, making it easier to apply existing norms to new and changed definitions.

5.3. The role of norms in RENISYS

In our approach, we use ontological definitions to ensure the *meaningfulness*, and normative definitions to guarantee the *acceptability* of specification changes.

Ontological definitions are represented in the form of type definitions. These definitions indicate the properties of the concept and its place of in the concept type hierarchy. For instance: "an editorial process is a type of workflow (supertype) in which a journal issue is produced (property)". State definitions instantiate these type definitions, e.g. "John is an (instance of) editor". Repeating the key definitions of action and composition norms:

action norm: a norm that describes the acceptable *operational* behaviour of some actor.
composition norm: a norm that describes the acceptable *specification* behaviour of some actor.

For the structure of the action and composition norms, Von Wright's list of *norm components* has been evaluated (this list, also used in MEASUR, is given in Fig. 4, with slightly adapted definitions).

From these components, the character, content and subject need to be included in each norm specification. The *character* is the deontic effect (permission, responsibility, or prohibition) of the norm. The *content* is an action or composition in action norms and composition norms, respectively. An *action* is some control process (i.e. an initiation, execution, or evaluation) applied to a workflow, a *composition* is a control process applied to a specification process (for example, the creation of a type definition). As Von Wright's *subject* in fact is the (generic) *actor* in RENISYS, it will from here on be referred to in that way.

The *condition* does not need to be represented explicitly. The norm applies if the user intending or expected to be performing an action or composition matches with the actor-part of the norm, and the intended or expected action or composition matches with the action or composition represented in the norm.

The *authority* capable of creating or changing the norm is given by the scope of the composition norms that apply to the defined norm (which together determine *who* can change it), and therefore does not need to be included in the norm being invoked. The *occasion* can also be left out, as all norms always apply to the whole network in the present time.

The basic structure of the RENISYS norms is as follows:

- *action norm*: deontic effect—actor—control process—workflow
- *composition norm*: deontic effect—actor—control process—specification process

Component	Description
Character	deontic effect (must/may/may not)
Content	prescribed action or activity
Condition	circumstance or state-of-affairs in which norm should be applied
Authority	agents who give or issue the norm
Subject	agents to whom norm applies
Occasion	location (space) or span (time) of norm

Fig. 4. *Von Wright's list of norm components (Liu, 1993).*

Process Level / Deontic Effect	Action Norm	Composition Norm
Permission	'a researcher *may submit* a paper'	'the scientific board *may create* an edit process *type*'
Responsibility	'an editor *must assess* a submission'	'a list owner *must change* the list registration *norm*'
Prohibition	'a reviewer *may not review* his own paper'	'a list member *may not modify* the list owner *status*'

Fig. 5. *Examples of RENISYS norm categories.*

There are three types of specification processes: *creations*, *modifications*, and *terminations* of knowledge definitions. As each combination of specification process and knowledge definition category requires a differently supported *definition process* in which the actual changes are made, there are twelve definition processes, one to 'create action norms', another one to 'modify state definitions', etc.

A formal representation and treatment of these norms and specification processes is given in de Moor (1999) and de Moor and Jeusfeld (2001) and is not relevant for the purpose of this paper. To illustrate the basic structure of the RENISYS norms, an informal example of each *norm category* (classified by the deontic effect and process level dimensions) is given in Fig. 5.

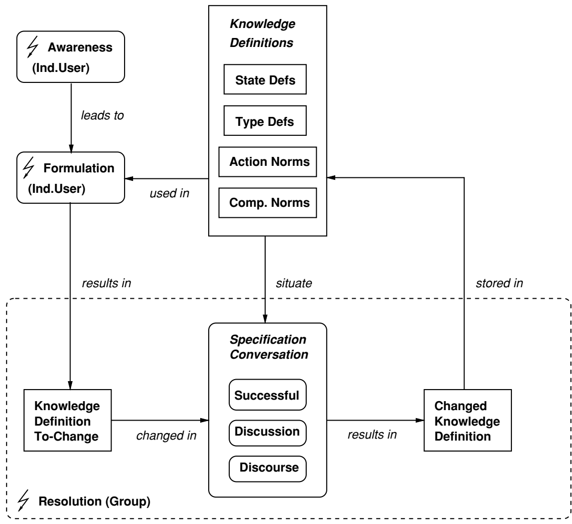

Fig. 6. *The situated conversation for specification model.*

6. RENISYS: Supporting Situated Conversations for Specification

We have now described the two main components of our framework for a context-based legitimate user-driven specification process: a *specification process model* and a normative *context model* which can be used to situate conversations for specifications. However, we have not yet explicitly integrated the two components into a complete framework for handling breakdowns. This model is outlined in Fig. 6. In this section, we briefly outline the method, and illustrate its use with an example.

6.1. An outline of RENISYS

The specification process starts with an individual user becoming *aware* of a breakdown. Presented with the existing knowledge definitions that are related to this breakdown, the user *formulates* the breakdown by identifying those definitions that need to be changed. In a group conversation for specification, the breakdown is *resolved* by legitimately changing each of these *problematic knowledge definitions*. The process in which a problematic knowledge definition is changed, is called the *active specification process*. For each composition of the active specification process, sets of *applicable composition norms* are calculated. A *composition* here is either the initiation, execution, or evaluation of the *active specification process*.

In de Moor (1999), the algorithms to do the complex conceptual graph calculations required to determine the applicable norm sets, are presented, involving generalization hierarchies of norm graphs. Here, it suffices to say that for each user and composition of the active specification process, a separate applicable norm set is calculated. The norms in these sets possibly have conflicting *deontic effects*, e.g. one saying that it is forbidden, another one that it is permitted for user John to execute the creation of a new review workflow (type) definition. In order to decide on which deontic effect applies, a norm conflict resolution mechanism is therefore also needed. Ours uses a variation of standard dynamic deontic logic (de Moor, 1999).

Using this mechanism, for each applicable norm set, its *resultant deontic effect* is calculated. This effect says if it is either permitted, required, or forbidden for a particular user to initiate, execute, or evaluate the particular specification process in which the problematic knowledge definition is to be changed. For instance, in the example, the resultant deontic effect would be that it is forbidden for John to execute review-type definition creation processes:

$$\mathbf{de_r}\left(D_{CN\_APPL(John,Exec\_Create\_Type(Review))}\right) = Forb$$

Once the resultant deontic effects for all norms sets have been calculated, the total sets of initiators I, executors X, and evaluators E for the current conversation for specification are known. For instance, John and Mary may be permitted to initiate the conversation, John is forbidden to execute it, while Mary and Jane are required to do so, etc. Using various techniques to support conversational moves in the spirit of Van Reijswoud's TPM, which are explained in detail in de Moor (1999), a rational discoursive specification process for the selected users in their prescribed conversational roles can now be enabled. In RENISYS, this is done by inviting selected users to log on to a web page that represents the particular conversation state in which the change process has arrived.

6.2. Using RENISYS: A typical example

A detailed analysis of the functionality of RENISYS does not fit within the scope of this paper and has been done elsewhere, as mentioned. Instead, we give an example of how the tool implementing RENISYS can be used. At the moment, only a prototype Web server is operational. Our intention is to expand the prototype into a robust, fully functional version that can be used to support realistic evolutionary processes. However, the plausibility of the method has already been demonstrated by successfully analyzing evolutionary problems and suggesting solutions in real cases of virtual professional communities. Cases analyzed include an electronic law journal, a global research network, and an electronic healthcare network (de Moor, 1999; de Moor and Jeusfeld, 2001; de Moor and Peterson, 2001).

6.2.1. Example: Defining a review process. A temporary research network has been formed to handle the publication of the proceedings of a prestigious conference. All the activities are to be done on-line. John, the conference coordinator, thinks it is essential that a review process is defined.

- To start the change process, John accesses the *Problem Awareness*-page of the RENISYS web server. When prompted, he indicates that his problem is related to the workflows of the community. On the *Problem Formulation*-page, he is presented with a list of the currently defined workflows. As no review-workflow has been defined yet, he indicates that a new workflow definition needs to be *created*. In a text-field, John informally describes why he thinks this workflow should be created by the community.
- The active specification process required to handle this problematic knowledge definition is a creation process of a workflow-type definition. RENISYS calculates the applicable norm sets, their resultant deontic effects, and the legitimate sets of initiators, executors, and evaluators (I,X,E) of this particular specification process.
- Once the persons who are to play the conversational roles have been calculated, RENISYS sends an e-mail to all *initiators* of this process. The e-mail contains the following information: the problem owner (John), the requested change process (creation of the review workflow type), the rationale (John's free text message), the people to which this e-mail was sent (the other initiators), and the request to access the RENISYS *Problem Resolution*-page. On this page, the initiators have to indicate whether the change request should be honoured. In the current implementation it is sufficient if one of the initiators does so. Alternatively, a requirement could be built in that *all* initiators have to agree.
- Upon acknowledgment that the request is okay, RENISYS sends a similar e-mail to all the *executors*. These meet face to face, discuss by e-mail and phone, and finally agree on the requested knowledge definition change. Part of the new knowledge definition that the group proposes, is that any submitted paper should be reviewed. Once completed, RENISYS invites the *evaluators* to see if the proposed definition changes are acceptable.
- John is one of the evaluators. Having some experience with publication editing, he thinks that it would be a mistake to have all submitted papers reviewed. In his opinion, there should first be a pre-selection process to determine whether a submitted paper fits within the scope of the conference. Via the discussion facilities of the specification tool, he therefore

criticizes the proposed definition of the paper review process. In a short free text-message, he explains his doubts. The group of executors discusses his comments informally, and one of its members subsequently replies to John's statement. The groups says that it understands his concern, but that it wants to prevent editors from discarding potentially good papers, so they have decided not to include a preselection process. John sees their point, and, on behalf of all evaluators, approves of the review definition, making it legitimate.

- However, Mary, an experienced conference editor (conference editors are currently not in the paper review-definition group of executors), still thinks that reviewing all submissions is not a good idea. Since the definition has already been approved, she decides to start a discourse (which any community member can always do). In it, she challenges the set of composition norms by which this group of executors of workflow type definition creation process has been selected. In this discourse process, she argues that it is not fair that conference editors are not included in these workflow change processes that affect their work so much.
- As Mary got quite positive responses from the discourse, she therefore requests that a new composition norm is defined that allow conference editors to take part in any workflow type change process.[2] RENISYS again invites this relevant user group to take part in the required conversation for specification. This group agrees, and creates the requested composition norm definition. From now on, the group of review process specifiers also includes the conference editors.
- In this re-organised group, Mary launches a new conversation for specification, this time to *modify* the current review process definitions. Now part of the relevant user group, the conference editors are able to convincingly present their points of view to the other executors. The whole group agrees that a paper preselection process is indeed needed, and it modifies the review process type definition accordingly. Subsequently, it requests the creation of a new workflow type: the paper pre-selection process, and so on.

The example has illustrated how RENISYS can be used to facilitate and improve the dynamics of community information systems evolution. By providing structured conversational facilities and selecting the users most relevant to a particular change request, change processes to the community and its information system become more focused and manageable. By providing a mechanism to efficiently spawn new conversations for specification from existing ones, it becomes easier for community members to identify the true causes of the breakdowns in their work. In all, RENISYS should lead to more community learning and construction, instead of mere technical information systems development.

7. Conclusion

In today's networked world, ever more virtual professional communities emerge in which people collaborate to do business, research, and so on. These communities form complex socio-technical systems, which are prone to considerable change. Change in these systems should take place in the form of legitimate user-driven specification processes, which ensure that any change is both meaningful and acceptable to the community as a whole.

To support these change processes, traditional information systems development methods such as workflow modelling methods are no longer adequate, as they focus on representation instead of interpretation of the socio-technical system. Therefore, new systems development methods grounded in the neo-humanist paradigm are needed. These methods allow for different subjectivist and conflicting views to be reconciled. Two branches of information science grounded in this paradigm are the language/action perspective (LAP) and organisational semiotics (OS).

Although specification methods based on either of these lines of thought have much to offer in realizing the neo-humanist ideals, they can achieve even more in combination. In this paper, we showed how the RENISYS method, which supports the legitimate user-driven specification process, has roots in both fields. It combines the conversational Transaction Process Model from the LAP-based DEMO, with the normative basis provided by OS-representative MEASUR. In this way, situated conversations for specification are possible that help to ensure the legitimacy of socio-technical system evolution in virtual professional communities. One important contribution of this work is thus that we have built a bridge between LAP and OS, fields that complement and need eachother.

We do not claim to have found the ultimate approach, as legitimate user-driven specification is only one way to fill in the neo-humanist paradigm, and RENISYS only one method to support this process. However, our method provides an opportunity to operationalize the neo-humanist paradigm. By using the method in further case analysis, and especially by implementing the approach in robust, widely usable specification tools, neo-humanist claims can be tested, leading to further theory validation and construction.

The RENISYS method and tool provide instruments to facilitate research into the evolution of virtual communities, so important in this rapidly changing and integrating world. By continuing to develop RENISYS, we hope to make a contribution to a world in which information systems, instead of being artificial constructs impeding social change, are catalysts for viable community evolution.

Notes

1. Note that we do not completely agree with the view that the organisation *is* the information system (Stamper, 2000). We admit that the organisational requirements should predominate, but we still see a need for a separate technological domain of analysis. In this way, mismatches between requirement specifications and enabling technological implementations can be more clearly identified.
2. Note that composition norms themselves are within the scope of other composition norms, thus permitting legitimate changes to the change capability of the community.

References

Auramäki E, Lyytinen K. On the success of speech acts and negotiating commitments. In: *Proceedings of the First International Workshop on Communication Modelling, the Language/Action Perspective (LAP'96)*, July 1–2, 1996: Oisterwijk, The Netherlands, 1996:1–12.

Brooks FP. *The Mythical Man-Month: Essays on Software Engineering*. Reading, MA: Addison-Wesley, anniversary edition, 1995.

De Michelis G, Grasso MA. Situating conversations within the language/action perspective: The Milan Conversation Model. In: Furuta R, Neuwirth C, eds. *CSCW '94*, ACM, 1994:89–100.

de Moor A. *Empowering communities: A method for the legitimate user-driven specification of network information systems*. Ph.D. thesis, Tilburg University, The Netherlands, 1999.

de Moor A, Jeusfeld MA. Making workflow change acceptable. *Requirements Engineering* 2001;6(2):75–96.

de Moor A, Peterson R. Facilitating the evolution of electronic healthcare networks: Framing the changing socio-technical system. *International Journal of Healthcare Technology and Management* 2001;3(5/6):366–385.

Dietz JLG. Modelling business processes for the purpose of redesign. In: *Business Process Re-Engineering: Information Systems Opportunities*. Amsterdam: North-Holland, 1994:233–242.

Engelbart D. Toward high-performance organizations: A strategic role for group-ware. Technical report, Bootstrap Institute, 1992.

Fitzpatrick G, Welsh J. Process support: Inflexible imposition or chaotic composition? *Interacting with Computers* 1995;7(2):167–180.

Hanseth O. Integrating information systems: The importance of contexts. In: Stamper et al. eds. Collaborative Work, Social Communications and Information Systems, IFIP, 1991:133–156.

Hirschheim R, Klein HK. Realizing emancipatory principles in information systems development: The case for ETHICS. *Management Information Systems Quarterly* 1994;18(1):83–109.

Hirschheim R, Klein H, Lyytinen K. *Information Systems Development and Data Modeling—Conceptual and Philosophical Foundations*. Cambridge, UK: Cambridge University Press, 1995.

Hollingshead AB, McGrath JE. Computer-assisted groups: A critical review of the empirical research. In: *Team Effectiveness and Decision Making in Organizations*. San Francisco: Jossey-Bass, 1995:46–78.

Kensing F, Winograd T. The language/action approach to design of computer-support for cooperative work: A preliminary study in work mapping. In: Stamper et al. eds. Collaborative Work, Social Communications and Information Systems, IFIP, 1991:311–331.

Khoshafian S, Buckiewicz M. *Introduction to Groupware, Workflow, and Workgroup Computing*. New York: John Wiley & Sons, 1995.

Koh I, Heng M. Users and designers as partners—Design method and tools for user participation and designer accountability within the design process. *Information Systems Journal* 1996;6:283–300.

Liu K. *Semiotics applied to information systems development*. Ph.D. thesis, University of Twente, 1993.

Medina-Mora R, Winograd T, Flores R, Flores F. The ActionWorkflow approach to workflow management technology. *The Information Society* 1993;9(4):391–404.

Nonaka I, Reinmoeller P, Senoo D. The 'ART' of knowledge: Systems to capitalize on market knowledge. *European Management Journal* 1998;16(6):673–684.

Peterson RR, Smits M, Spanjers R. Exploring IT-enabled network organisations in healthcare: Emerging practices and phases of development. In: *Proc. of the 8th European Conference on Information Systems*, Vienna, 2000:1253–1260.

Schäl T. *Workflow Management Systems for Process Organizations*. Berlin: Springer Verlag, 1996.

Scheer A-W. ARIS. In: Bernus P, Mertins K, Schmidt G, eds. *Handbook on Architectures of Information Systems*. Berlin: Springer-Verlag, 1998:541–565.

Searle JR. *Speech Acts—An Essay in the Philosophy of Language*. Cambridge, UK: Cambridge University Press, 1969.

Stamper R. Language and computing in organised behaviour. In: van de Riet RP, Meersman RA, eds. *Linguistic Instruments in Knowledge Engineering*. Amsterdam: Elsevier Science Publishers B.V., 1992:143–163.

Stamper R. Social norms in requirements analysis—An outline of MEASUR. In: *Requirements Engineering: Technical and Social Aspects*. Academic Press, 1994:107–139.

Stamper R. Signs, information, norms, and systems. In: Holmqvist B, Andersen PB, eds. *Signs at Work*. Berlin: De Gruyter, 1996.

Stamper R. New directions for systems analysis and design. In: Filipe J, ed. *Enterprise Information Systems*. Boston: Kluwer Academic Publishers, 2000:14–39.

Stamper R. Extending LAP and focusing OS. In Liu K, ed. *IFIP WG8.1 Working Conference—Organizational Semiotics: Evolving a Science of Information Systems*, July 23–25, 2001, Montreal.

Stamper RK, Kerola P, Lee R, Lytinen K, eds. *Collaborative Work, Social Communications and Information Systems*. IFIP, 1991.

Suchman L. Do categories have politics? The language/action perspective reconsidered. *Computer Supported Cooperative Work* 1994;3(2):177–190.

Talbott S. *The Future Does Not Compute: Transcending the Machines in Our Midst*. O'Reilly & Associates, 1995.

Taylor JR. The limits of rationality in communication modeling: A socio-semiotic reinterpretation of the concept of 'speech act'. In: *Proceedings of the Third International Workshop on Communication Modelling, the Language/Action Perspective (LAP'98)*, June 25–26, 1998, Steningevik, Sweden, 1998:35–46.

van Reijswoud V. *The structure of business communication: Theory, model and application*. Ph.D. thesis, Delft University, 1996.

Verharen E. *A language-action perspective on the design of cooperative information agents*. Ph.D. thesis, Infolab, Tilburg University, 1997.

Wand Y, Monarchi DE, Parsons J, Woo CC. Theoretical foundations for conceptual modelling in information systems development. *Decision Support Systems* 1995;15(4):285–304.

Weigand H, Dignum F. Formalization and rationalization of communication. In: *Proceedings of the Second International Workshop on Communication Modelling, the Language/Action Perspective (LAP'97)*, Veldhoven, June 9–10, 1997, The Netherlands, 1997:71–86.

White SK. *The Recent Work of Jürgen Habermas: Reason, Justice, and Modernity*. Cambridge, UK: Cambridge University Press, 1988.

Winograd T. A language/action perspective on the design of cooperative work, report no. CSLI-87-98. Technical report, Center for the Study of Language and Information, Stanford University, May 1987.

Winograd T, Flores F. *Understanding Computers and Cognition—A New Foundation for Design*. Ablex Publishing Corporation, 1986.

Aldo de Moor is an assistant professor at the Department of Information Systems and Management of Tilburg University. He holds a Ph.D. in Information Management and has been a visiting researcher at, among other institutions, the University of Guelph, Canada, and the University of Technology, Sydney. His research interests include the evolution of virtual communities, communicative action theories, user participation in systems development, normative system specification methods, and the impact of information and communication technologies on society. Projects in which he has participated include the development of an electronic journal, a business negotiation support system, and various virtual community and network development projects.