View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Research Papers in Economics

The Initialization of Conversations for Specification: A Context of Social Norms

Aldo de Moor

Infolab, Tilburg University, The Netherlands
ademoor@kub.nl

Abstract

Virtual professional communities require a legitimate user-driven specification approach of their network information systems. The specification changes produced should be legitimate in the sense that they are not only meaningful but also acceptable to all members of the community. We regard specification processes as conversations for specification. A Specification Process Model is presented that is grounded in the theory of communicative action, and builds upon DEMO and the rational discourse represented in the Transaction Process Model. To initialize the conversations for specification, social norms play an important role. The RENISYS method is introduced which supports such contextualized conversations for specification.

1 Introduction

Collaborative work is increasingly being done in a distributed fashion, supported by commonly available Internet-based information tools such as mailing lists or web applications. We define the *virtual professional communities* in which such collaboration is to take place as communities or networks of professionals whose

The copyright of this paper belongs to the paper's authors. Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage.

Proceedings of the Fifth International Workshop on the Language-Action Perspective on Communication Modelling (LAP 2000)
Aachen, Germany, September 14-16, 2000
(M. Schoop, C. Quix, eds.)
http://www-i5.informatik.rwth-aachen.de/conf/lap2000/

collaboration on activities required to realize shared goals is mostly or completely computer-enabled. The workflows of these communities are often supported by network information systems consisting of linked and configured standard information tools. The communal requirements and enabling information technologies typically evolve strongly, while the users ought to have an important role both as sources and as modellers of the system specifications. Active user participation in the specification process of such continuously evolving socio-technical network information systems is very important, since community members have the most detailed knowledge about when breakdowns in work arise and how they can be resolved [3]. To increase the efficiency and willingness of users to participate in system change processes, it must be known exactly for each specification process what users are to take part, and in which roles they are to be involved. To adequately determine the *relevant user group*, a *legitimate user-driven specification approach* is required. First, in such an approach, the community members are not just to provide specification knowledge, but also to control themselves the process in which this knowledge is actually produced, from the start of a change process to its conclusion, thus making the specification process truly *user-driven*. Second, these change processes must be *legitimate*, in the sense that any change that is implemented is semantically (and pragmatically) meaningful and also acceptable to all members of the community.
The rationale, characteristics of, and support for the legitimate user-driven specification process are described in detail in [4, 5], but are not our focus here. In this paper, we summarize how our approach, the RENISYS (**RE**search **N**etwork **I**nformation **SY**stem **S**pecification) method is grounded in the language/action perspective by facilitating conversations for specification. It is then shown how these conversations are to be initialized in a context of representations of some of the social norms grounded in the community.
In Sect. 2, we use speech act theory to describe specification processes in professional communities, by viewing them as series of related conversations. Although it provides an important starting point for specification process support, the conversational model in itself is insufficiently capable of handling rational discourse, because it does not pay attention to how to deal with questioning background assumptions. In Sect. 3, a Specification Process Model is built, drawing upon Habermas's theory of communicative action, Dietz's DEMO specification method, and Van Reijswoud's Transaction Process Model. A remaining problem, discussed in Sect. 4, is how contextual knowledge is to initialize these conversations. Our solution is to use representations of the social norms that guide the behaviour of a community, called composition norms, to link context and conversations. The context-based specification process model that makes use of these composition norms is introduced in Sect. 5. This model forms the basis for our RENISYS approach.

2 Conversations for Specification

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 [7]. 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 type of article submission process. Evidence for the effectiveness of predefined conversation models is ambiguous [1]. 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 [26]. This kind of conversation is the basis for the well-known Coordinator and ActionWorkflow modelling methods [10]. Fig. 1 represents a conversation for action as a state transition network [27, 26].

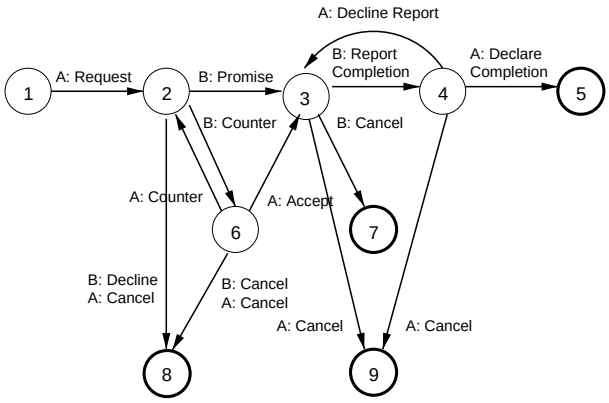

Figure 1: A State Transition Diagram of a Conversation for Action [26]

Many different types of conversations, but especially the conversation for action, can play some role in the specification process. We will illustrate the ideas presented in this paper by taking examples from specification processes related to an electronic journal publication process. We assume there is an electronic 'LAP-Journal', of which John is an author and Jane the editor.

The system specification process is triggered by breakdowns in work, in the sense of [27]. For example, a breakdown experienced by John could be that he finds the

editorial process of his submitted article taking too long.

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 [26]. To do so, a conversation framework is needed that combines specialized as well as more general conversation patterns that can provide support for breakdown-initiated conversations [9]. 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*.

3 Constructing a Specification Process Model

An important limitation of Searle's original speech act theory is that it stresses the conversational role of the *speaker*, while ignoring the role the *hearer* plays in the success of speech acts. This makes it hard to know whether a hearer does something because he, fully informed and unpressured, accepts the speech act made by the hearer, or because he is, for example, insufficiently knowledgeable or forced in some way to accomodate the speaker. In virtual professional communities this would be a most undesirable situation, as legitimacy of the changes produced in conversations for specification is of the greatest importance. These communities require information system development methods that do take the hearer into account as well by facilitating rational discourse. Habermas's theory of communicative action provides a comprehensive conceptual framework in which to ground such methods that focus on the role of the hearer as well.

Sect. 3.1 gives our view on the theory of communicative action. One specification method which is partially based on this theory is the DEMO method. It is contrasted with RENISYS in Sect. 3.2. In Sect. 3.3, it is argued that there remains a need for conversation protocols that prescribe the conversational moves that actors are allowed to make. Van Reijswoud's Transaction Process Model is presented, which combines DEMO with the theory of communicative action to model the various conversational moves that occur in rational discourse. In Sect. 3.4, this model forms the basis for the construction of our own Specification Process Model, which is used to model conversations for specification.

3.1 The Theory of Communicative Action

In Habermas's perspective, actors who are willing to coordinate their actions consensually aim to achieve an understanding through a "cooperative process of interpretation aimed at attaining *intersubjectively recognized definitions of situations*[1]"

[1] My emphasis.

[25, p.39]. In our approach, such definitions are the result of conversations for specification.

Specification changes in virtual professional communities, by necessity are often compromises, because of the different interests represented and values and knowledge possessed by the various community members involved. Concretely, legitimacy is enforced by demanding that the procedures for compromise construction themselves are justified, or at least justifiable, through discourse. General criteria to be used in the construction of such procedures include procedural equality, participation, and non-manipulation. Based on these criteria, Habermas developed his *rules of discourse*, such as "each subject is allowed to introduce any proposal into the discourse". A summary of the rules is given in [25, p.56].

These rules are in line with the requirements of the specification discourse taking place in virtual professional communities. In this discourse, all validity claims should be open for discussion, since only agreements that are intersubjectively valid on all dimensions can ensure continued participation (ibid.,p.40) A basic tenet of the theory of communicative action is that ordinary language competence suffices to use the whole system of interrelated validity claims for the coordination of action (ibid.,p.39) . Thus, the users should be given the freedom to discuss (combinations of) validity claims, while they themselves decide which claims to discuss and how to discuss them. In this way, no formal representation of validity claims and their dependencies is necessary, nor desirable. Still, formally modelling the *conversational roles* that users play in the discourse process is most helpful in creating the right *preconditions* for successful informal discourse.

The theory of communicative action has its limitations as well. Habermas's classification of speech acts is useful for structuring dyadic communicative acts between speaker and hearer, but does not allow for handling larger communication contexts (including conversations) [1]. Furthermore, his category of regulativa, comprising both Searle's directives and commissives, is too broad to be useful in system development practice. For example, one often needs to know whether somebody is commanding (directive) or promising (commisive) something [1]. Therefore, we partially follow Verharen [23], who adopts the Searlean classification of speech acts, extended with Habermas's validity claims attached to the illocutionary acts. Contrary to Verharen's focus on claims to power, authority, and charity, ours is on the claim to justice, which we rename into a *claim to legitimacy*. Although both concepts have fairness and acceptability as their key elements, we prefer to use the latter. The term 'justice' has heavy connotations concerning the idea of protecting *against* wrong-doing, whereas 'legitimacy' has more of a positive, *constructive* flavour.

Summarizing, what we need is a natural language-like discussion facility based on a formal communicative action-grounded discourse coordination mechanism.

This provides us with a universal approach for facilitating legitimate user-driven specification, which is independent of the specific usage context in which it is taking place. Such a coordination mechanism should incorporate the rules of discourse, which help to approximate Habermas's 'ideal speech situation', in which discursive equality, freedom, and fair play are guaranteed.

However, and this is crucial: Habermas does not provide full prescriptions for how these procedures are to be interpreted. Instead, his rules of discourse need to be interpreted and operationalized by the actors being affected by them [25, p.76-77]. Thus, we need (1) to operationalize the conversations for specification while incorporating the rules of discourse and (2) find a way to select the relevant actors affected by a particular conversation. The first we do in the remainder of this section, the selection of the relevant actors is discussed in Sect. 4.

3.2 DEMO: a LAP-Oriented Specification Method

To understand how the language/action perspective can contribute to systems specification, Dietz's DEMO (Dynamic Essential Modelling of Organisations)-method is briefly reviewed. DEMO is a cross-disciplinary theory about the dynamics of organizations, as well as an organizational and information systems analysis method. It is based on both Searlean and Habermasian theory [7, 1]. DEMO was contrasted with the RENISYS method in [20]. Two main ideas of DEMO are adopted in RENISYS. First, in DEMO a distinction is made between actors and subjects. An *actor* is an entity defined by the set of actions and communications it is able to perform, and is realized by subjects. A *subject* refers to a particular person in some functional role. Second, the idea of a *transaction* as a basic unit of communication is used (Fig. 2). This is the core modelling concept of DEMO. Hierachically, a transaction stands between an individual speech act and a conversation. A transaction consists of three stages: the actagenic, action, and factagenic stage. During the *actagenic phase*, initiated by actor A, agreement is reached between actor A and actor B in an actagenic conversation about the future execution of an action by actor B. During the *action phase* this essential action is executed by actor B. In the *factagenic phase* actor A and B reach agreement in a factagenic conversation about the facts that have been accomplished as a result of the execution by actor B. Actor A is called the initiator of the transaction and actor B the executor. The behaviour of an organization is conceived of as the carrying out of a set of transactions. Every (essential) action is embedded in a transaction, and every established fact is the result of the successful completion of a transaction.

In RENISYS, a transaction is similar to a workflow at the operational level, and to a specification process at the specification level. However, there are also some differences between both approaches. First, in DEMO, an actor is often known as a process name, e.g. 'Planning Production' or 'Shipping', while a subject is iden-

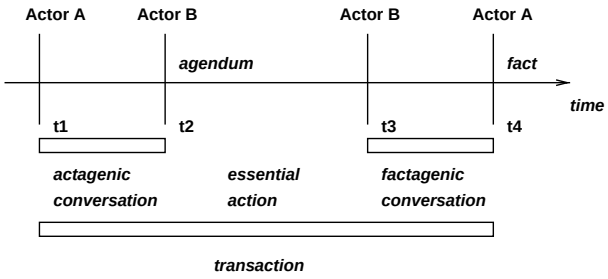

Figure 2: The DEMO Transaction

tified by a functional role, e.g. 'Logistics Manager' [21]. In RENISYS, however, the functional role is expressed by the actor entity, whereas a subject refers to an individual person. Second, a transaction in DEMO only has an initiator and an executor for conversation roles. The initiator has to approve of the result of the action performed by the executor. In RENISYS, however, a separate *evaluator* role is distinguished for this purpose. Third, RENISYS provides guidance for the specification process by means of its reference framework, in which various kinds of specification dependencies are represented. This framework links conceptual entities from the problem domain (goals and activities) via the human network (organizational structures) to the information system domain (information and communication processes and tools). Finally, DEMO does not take into account the existing information system when producing new specifications, it is not completely clear how the DEMO models are to be used in system design, and there is a lack of deontic or normative concepts [23]. RENISYS, however, provides explicit support for the evolution of analysis and design specifications by indicating which actors can legitimately make what specification changes, and who is to implement them.

3.3 The Transaction Process Model

It is often a problem for the hearer to classify the illocutionary force 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 [27, 13]. In addition to the modelling techniques such as used in the conversation for action approach (see Fig. 1), theory-grounded *conversation protocols* are therefore needed that 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) [22].

The TPM, itself based on DEMO, 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 the transaction concept. The model is represented as a state transition diagram, similar to Fig. 1, 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. Of course, they are embedded in this process, but the acts themselves are aimed at the *production*, not at the *planning* or *validation* of the results.

Whereas the state transition technique is generally applied to modelling the behaviour of objects in the object world, in the TPM it is used to represent the communication behaviour of subjects in the intersubject world. Besides being able to model successful communication processes, the TPM also allows for the representation of discussion and discourse, as proposed in the theory of communicative action. The model 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. The *discourse-layer* contains discourse with the purpose of restoring background conditions. 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.

3.4 The Specification Process Model

The TPM can be applied in different ways. In [18], it is used to model actual business conversations, and thus needs to deal with such complexities as the distinction between strategic and communicative action and the reformulation of unclear statements. Similarly, Steuten makes an extensive study of the roles discourse analysis and conversational analysis can play in providing a foundation for the modelling of business conversations. Grounded in empirical data, both of her analysis approaches are concerned with the way in which coherence and sequential organization of real world-conversations are produced and comprehended [17]. However, this complexity is not needed in our case, as we only need to *generate* relatively simple potential conversational moves instead of to *interpret* complex natural language statements.
Although the TPM forms the basis for the modelling of 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. 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 [4], but are less relevant here.
In the SPM, we formalize conversations as little as possible, in order to provide flexibility and not to cognitively overburden users. Thus, although a user can start a discussion to, say, question the sincerity of another user's communication act, the initiator does not need to formally indicate why he does so. The reason for this is that RENISYS *enforces* the legitimacy of specification processes by only inviting those participants to take part in some conversation for specification who are justified to do so. Once they have been selected, they are free to discuss in the way they like. This is a new application of the TPM: whereas the latter is a model to *represent* conversations, RENISYS uses its SPM to *select* participants who are to take part in them. To illustrate the use of the SPM, we show the representation of a *successful specification process* to modify the editorial process type definition, as suggested by author John in the previous example. This representation is similar to those given in [22, p.95], showing four of the in total 23 conversation acts. A complete overview of all conversation acts making up the SPM is given in App.A of [4]. For each of these acts, it is discussed there which conversational roles can perform as speakers and hearers of the act. In this paper, we do not need to show them all, since we focus here on how to initialize the conversational roles, not on the actual conversation process in which these roles take part.

In the example, the sequence of conversation acts and definition processes is the following:

Act	*Description*	Resulting State
CA_1:	I: C_{legit}[propose(directive)<mod_type_def(edit),now>]	Directed
CA_2:	X: C_{legit}[promise(commissive)<mod_type_def(edit),now>]	Committed
DP:	X: DP_{legit}[define(execute)<mod_type_def(edit),now>]	Executed
CA_3:	X: C_{legit}[rep_compl(decl.)<mod_type_def(edit),now>]	Decl.(Completion)
CA_4:	E: C_{legit}[decl._success(decl.)<mod_type_def(edit),now>]	Decl.(Success)

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

Here, the (legitimate) initiator asks the executor(s) to modify the existing edit process type definition. The executor (which can consist of more than one person) promises to do this, and later presents a modified editorial process definition to the evaluator(s), who, in this case, approves the proposed change.

4 Contextualizing the Conversation

One major criticism of the application of speech act theory in systems development is that it is not able to represent what people really do, as it is said to provide models

that are too rigid and simplistic to capture the complexities of actual work practices [1, 23].

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 [26, 2, 19]. Thus, a fundamental problem has not been addressed by the TPM (and by the SPM, so far): how to make the link between the specific 'social/organizational and work situations' and the conversations for specification? *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 specification conversation is taken into account [2]. However, the idea of context in speech act theory is still relatively unexplored. Although it is recognized that each individual speech act is embedded in an abstract social context (e.g. conditions such as the propositional content and preparatory rules and the fact that the lifeworlds of communicants must be compatible), it is not yet very clear how exactly context is to be used in LAP-grounded systems development [23]. A major implication of the language/action perspective is that context interpretation cannot be fully automated, but to a large extent remains to be done by persons [8, 24]. These ideas are elaborated upon next. First, we describe the CHAOS system, a related approach that also strongly emphasizes the role of context in conversations, then we present our own conversation context model, after which we discuss the role of norms in our approach.

4.1 Related Work: The CHAOS System

A sophisticated approach to making the link between speech act-based conversations and context, the CHAOS system, is proposed by Simone and Divitini (1997). They distinguish four different types of context: operation, communication, organization, and linguistic contexts. In their view, communication in cooperative groups can become problematic because it is often difficult to keep the contexts of the cooperative work up-to-date and consistent.

Simone and Divitini claim that the LAP can be helpful to identify the events that characterize the evolution of the contexts in which communication occurs and takes its meaning. Similar to our approach, their CHAOS system does not let the computer make all required specification knowledge inferences. Instead, it provides the user with relevant information that is as much as possible contextualized by the communication processes that generated it, after which they leave it up to the user make the appropriate inferences. To accomplish this, a user model is applied which is not fixed a priori, but gradually defined by the conversations and commitments in which the user is involved. Using ideas like *ontological levels* (e.g. an entity is 'un-

der discussion' or 'existing'), *commitment life cycle* (e.g. a commitment is 'taken' or 'done'), and the *partiality of shared knowledge* (knowledge is either 'consciously shared knowledge', 'shared knowledge' of which users are unaware that they share it, and 'private knowledge'), mechanisms have been developed that allow possible inconsistencies among the knowledge of group members to be discovered. These mechanisms allow for the most appropriate context, depending on a particular user's needs, to be found.

Although both CHAOS and RENISYS acknowledge the important role that contexts play in the conversation process, this role is a very different one. In both approaches, only the relevant set of people is to be involved in the sharing (and creating) of organizational knowledge. However, the main focus of CHAOS is on dealing with the effects of the partiality of knowledge sharing, instead of providing full support for communicative action, which is the main concern of the SPM in RENISYS. In the CHAOS approach, the context of the commitment at the time of its definition can be reconstructed. It remains the responsibility of an individual user with a pending commitment to determine if this context is still valid. If not, then the evolution of the knowledge objects can be presented to the user, including the (partial) views that other users may have on them. Although the approach does have the notion of 'privileged users' having the authority to provide information about the status of objects and actions for which they are responsible, there is no mechanism for ensuring the legitimacy of specification changes. It is left up to the individual users interpreting contexts during commitment definition, not to communal norms, to determine who else to involve.

On the other hand, RENISYS does not need to incorporate complex partial knowledge sharing mechanisms to deal with ambiguities. It assumes knowledge definitions to have global scope as they can only be made by legitimate actors. This scope does not mean that every actor needs to be aware of the current status of all entities. The specification mechanism of RENISYS handles breakdowns by focusing on the identification of the relevant actors to be involved in specification discourse. Once a conversation for specification has started, the related definitions (i.e. definitions of concepts used in the definition under debate) can be presented and informally discussed. We assume that such an informal group discussion allows for a clear identification of the meaningfulness and acceptability of the definitions, while at the same time foregoing much of the cognitive overhead required by knowledge representation-intensive approaches like CHAOS.

4.2 The RENISYS Conversation Context Model

We now define the meaning of context in our approach (Fig. 3). It consists of two parts, the external and the internal conversation context.

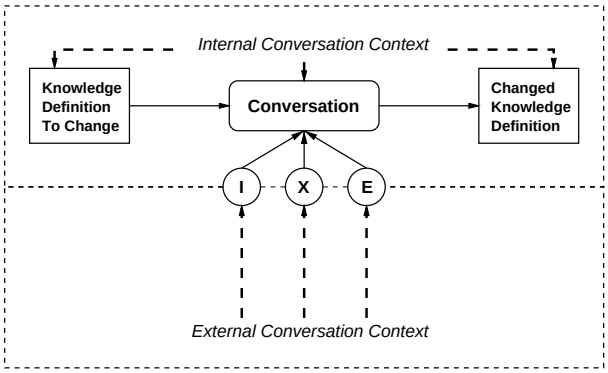

Figure 3: Contexts and Conversations

Definition 1

- **internal conversation context:** the knowledge definitions which are 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.

□

The internal conversation context gives meaning to the definition being changed, as it situates the definition in a web of semantically related definitions, that are already meaningful to and accepted by the community. For example, the definition of the editorial process may include links to the submission and review workflows, and to submitted and edited papers as input and output objects, respectively. Each of these concepts in turn have their own definition. The external conversation context, which is used to determine the relevant users to involve in definition change processes, is related 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 [19].

In order to model internal and external conversation contexts, different kinds of *specification knowledge categories* are necessary. We distinguish four such categories [6, 4]: *type definitions* determine the ontological meaning of concepts (e.g. a mailing list is an information tool that resends received mails), while *state definitions* capture states-of-affairs (e.g. Jane is the list owner of the LAP-Journal mailing list). Furthermore, there are two kinds of norms, *action norms* and *composition norms*, which regulate operational and specification behaviour, respectively. We describe the properties of these norms later in this paper.

4.3 From Power to Norm-Based Authority

Conversations for specification, taking place in a basically egalitarian virtual professional community, are a prime example of a democratic 'open discussion' approach necessary to define problems and solutions [28]. However, democracy is more than just discussion. Ultimately, somebody must be in a position of *authority* to coordinate the discussion and to make decisions. There are two ways for an actor to obtain such authority. The first one is based on *power*, as a result of structures of domination. Here, an actor has command over either persons or objects. The second way is by means of *norms* that actors use to sanction their own conduct and that of others. As virtual professional communities do not have pre-existing structures of domination, our focus is on norms as sources of authority.

A general definition of a norm is a principle of right action binding upon the members of a group and serving to guide, control, or regulate proper and acceptable behaviour[2]. This definition states that a norm is an instrument that can be used to identify which behaviour of a particular actor is just from the point of view of the other members of the group to which the norm applies. Norms can act both as affordances and as constraints: on the one hand they describe socially acceptable behaviour, on the other hand they can also be used to identify behaviour that is not allowed. Norms, by permitting and constraining actions, can thus be used to guide the work of groups and evolutionary system change [15, 28].

Crucial to successful network information system development is the continuous *evolution* of the *norm-based authority* structures. Contrary to a power-based authority, which cannot easily be challenged, norm-based authority should be a relative one, which can always be questioned and changed. There are two reasons for addressing authority issues gradually rather than radically: first, the communication within a group continuously grows in size and diversity, and second, the group participants require time to understand the implications of the mechanisms developed [12]. As the evolution of authority structures in network communities is grounded in a unique social context, an important question is how this work and organizational context can be linked to the process of changing the norms, in such a way that these norms continue to reflect the interests and social values of the various participants. Current system development approaches, however, pay little attention to the process in which the norms that govern these systems are being established [23]. However, such sophisticated norm definition mechanisms, grounded in the context of work, are needed for a virtual professional community to (continue) to function.

[2]Webster's Dictionary:`http://www.m-w.com/dictionary.htm`

4.4 Norms in RENISYS

The legitimate user-driven specification process was said to result in specification changes that are both meaningful and acceptable to the members of a virtual professional community. In our approach, we use ontological definitions to ensure the *meaningfulness*, and normative definitions to guarantee the *acceptability* of specification changes, distinguishing both action and composition norms:

Definition 2

- **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.

□

Thus, action norms regulate the work that the community is to carry out, whereas composition norms define who is to be involved in the changing of the socio-technical system in which this work is done, including changes in the composition norms themselves.

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

Figure 4: Examples of RENISYS Norm Categories

To illustrate the basic structure of the RENISYS norms, an informal example, drawn from the journal publishing domain, of each *norm category* (classified by the deontic effect and the operational versus specification process level dimensions) is given in Fig. 4.

The use of norms in RENISYS is partially inspired by the MEASUR research programme, based on Stamper's semiotic theory of systems development [15]. Elements adopted in RENISYS are the distinction between ontological and normative knowledge, the deontic effect classification of norms, and the recognition of informal norms. There are some important differences as well, though. RENISYS does not support the complete traditional information system development process as

MEASUR does. Instead, we have adopted, simplified, and extended some of its core ideas, notably on ontologies and norms that are useful for our particular purpose of legitimate user-driven specification. Differences include: the action/composition norm classification and a different subdivision and representation of ontological and normative knowledge [4].

5 RENISYS: Supporting Contextualized 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 *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. 5.

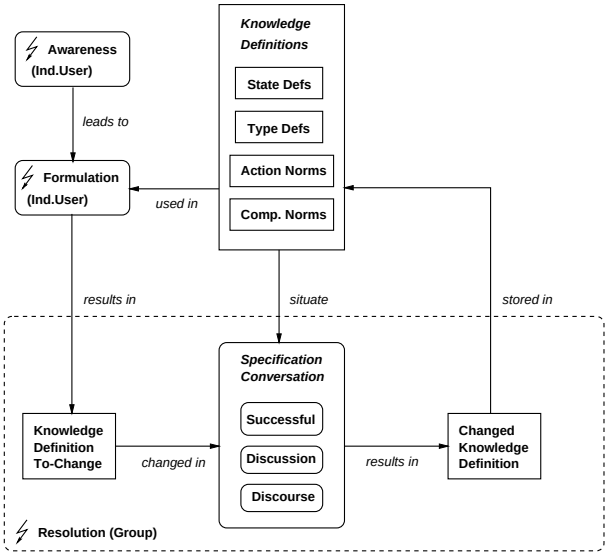

Figure 5: A Context-Based Conversation for Specification Model

The specification process starts with an individual user becoming *aware* of a breakdown in his work or supporting information tools. In the previous example, author John was unsatisfied with the slow response to his submitted paper. 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 the example, John suggested that the editorial workflow process is to be redefined. The breakdown is *resolved* by the executors legitimately changing each such a *problematic knowledge definition* by an appropriate conversation for

specification.

For example, the editorial process type definition may be modified to include an 'acknowledge submission received' process. To find out which users to involve in the initiation, execution, and evaluation of the specification process, for each of these definitions, a set of *applicable composition norms* is calculated (for each combination of user and composition). A *composition* is either the *initiation*, *execution*, or *evaluation* of the *active specification process* in which the problematic knowledge definition is to be changed. There are three types of specification processes: *creations*, *modifications*, and *terminations* of knowledge definitions. As each combination of specification process and knowledge category requires a differently supported *definition process* in which the actual changes are made, there are twelve definition processes [4]: one to 'create action norms', another one to 'modify state definitions', etc. Again, we do not focus on the procedural aspects of conversations for specification in this paper, but on their initialization. The dynamics of these definition processes are therefore not discussed here.

In [5], 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 (i.e. user Jane and composition 'execution of editorial process type definition modification') of the active specification process, a separate applicable norm set is calculated. The norms in these sets can have conflicting *deontic effects*, e.g. one saying that it is forbidden, another one that it is permitted for user Jane to execute the modification of type definitions of the edit process. To handle these conflicts, a norm conflict resolution mechanism is needed. Ours uses a slight variation of standard dynamic deontic logic [11]. An example of how conflicting norms could apply to the same user is that Jane, who is the journal editor has these two norms in her applicable norm set for the execution of the edit type modification process: a general norm saying that it is forbidden for editors to change their own editorial processes, as well as a more specific norm saying that it is permitted for *journal*-editors to be involved in the changing of such workflows.

For each applicable norm set, its *resultant deontic effect* is calculated, which says if it is permitted, required, or forbidden for a particular user to either initiate, execute, or evaluate the particular specification process in which the problematic knowledge definition is to be changed. In the example, the resultant deontic effect would be that it is forbidden for Jane to modify edit-type definition processes, since prohibitions have precedence over privileges in our logic:

$$\mathbf{de_r}(D_{CN\_APPL(Jane,Exec\_Modify\_Type(Edit))}) = Forb$$

Once the resultant deontic effects for all applicable norms sets have been calcu-

lated, the sets of initiators I, executors X, and evaluators E for the current conversation for specification are known (see Fig. 3). Using various techniques to support conversational moves in the spirit of Van Reijswoud's TPM, which are explained in detail in [4], a rational discoursive specification process for the selected users in their prescribed conversational roles can now be enabled. These procedural aspects, however, deserve a detailed explanation for which space is lacking in the current paper, which limited itself to sketching how these procedures can be initialized.

6 Conclusion

In this paper, we focused on how to initialize the conversational roles of the conversations for specification which are necessary for the legitimate user-driven specification of network information systems. The conversations themselves are based on a rational discoursive Specification Process Model. The initialization of the conversational roles (the initiators, executors, and evaluators of conversations), on the other hand, is dependent on the composition norms that define the specification behaviour of community members. The RENISYS specification method operationalizes this legitimate user-driven approach. It has been implemented in a prototype web-based tool, which we intend to upgrade to a robust version in the near future. Then, we plan to refine and extend theory and method using the tool in a range of realistic cases.

Acknowledgements

The author wishes to thank the reviewers for their constructive feedback.

References

[1] E. Auramäki and K. Lyytinen. 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), Oisterwijk, The Netherlands, July 1-2, 1996*, pages 1–12, 1996.

[2] G. De Michelis and M.A. Grasso. Situating conversations within the language/action perspective: The Milan Conversation Model. In R. Furuta and C. Neuwirth, editors, *CSCW '94*, pages 89–100. ACM, 1994.

[3] A. De Moor. Toward a more structured use of information technology in the research community. *The American Sociologist*, 27(1):91–101, 1996.

[4] A. De Moor. *Empowering the User: A Method for the Legitimate User-Driven Specification of Network Information Systems.* PhD thesis, Tilburg University, The Netherlands, 1999. ISBN 90-5668-055-2.

[5] A. De Moor. Composition norm dynamics calculation with conceptual graphs. In *Proceedings of the Eighth International Conference on Conceptual Structures, ICCS2000, Darmstadt, Germany, August 14–18, 2000*, 2000.

[6] A. De Moor and G. Mineau. Handling specification knowledge evolution using context lattices. In *Proceedings of the Sixth International Conference on Conceptual Structures, ICCS'98, Montpellier, France, August 10–12, 1998*, pages 416–430, 1998.

[7] J.L.G. Dietz. Modelling business processes for the purpose of redesign. In *Business Process Re-Engineering: Information Systems Opportunities*, pages 233–242. North-Holland, 1994.

[8] O. Hanseth. Integrating information systems: The importance of contexts. In Stamper et al. [16], pages 133–156.

[9] F. Kensing and T. Winograd. The language/action approach to design of computer-support for cooperative work: A preliminary study in work mapping. In Stamper et al. [16], pages 311–331.

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

[11] J.-J.Ch Meyer and R. Wieringa, editors. *Deontic Logic in Computer Science: Normative System Specification*. John Wiley & Sons Ltd., 1993.

[12] V.L. O'Day, D.G. Bobrow, and M. Shirley. The social-technical design circle. In *Proceedings of the Conference on Computer Supported Cooperative Work, Cambridge, MA, USA, 1996*, pages 160–169, 1996.

[13] T. Schäl. *Workflow Management Systems for Process Organizations*. Springer Verlag, 1996.

[14] C. Simone and M. Divitini. LAP-based mechanisms for maintaining the contexts of cooperation. In *Proceedings of the Second International Workshop on Communication Modelling, the Language/Action Perspective (LAP'97), Veldhoven, The Netherlands, June 9-10, 1997*, pages 105–116, 1997.

[15] R. Stamper. Language and computing in organised behaviour. In R.P. van de Riet and R.A. Meersman, editors, *Linguistic Instruments in Knowledge Engineering*, pages 143–163. Elsevier Science Publishers B.V., 1992.

[16] R.K. Stamper, P. Kerola, R. Lee, and K. Lytinen, editors. *Collaborative Work, Social Communications and Information Systems*. IFIP, 1991.

[17] A. Steuten. *A Contribution to the Linguistic Analysis of Business Conversations within the Language/Action Perspective*. PhD thesis, Delft University, 1998.

[18] A. Steuten and V. Van Reijswoud. The interpretation of business communication: The application of functional grammar and the transaction process model. In *Proceedings of the First International Workshop on Communication Modelling, the Language/Action Perspective (LAP'96), Oisterwijk, The Netherlands, July 1-2, 1996*, pages 77–91, 1996.

[19] J.R. Taylor. 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), Steningevik, Sweden, June 25-26, 1998*, pages 35–46, 1998.

[20] N. Van der Rijst and A. De Moor. The development of reference models for the RENISYS specification method. In J.F. Nunamaker Jr. and R.H. Sprague Jr., editors, *Proceedings of the 29th Hawaii International Conference on System Sciences, Maui, January 3–6, 1996*, pages 455–464. IEEE Computer Society Press, 1996.

[21] N. Van der Rijst and V. Van Reijswoud. Comparing speech act based modeling approaches for the purpose of information system development. In *Proceedings of the 3rd European Conference on Information Systems, Athens, June 1-3,1995*, pages 353–365, 1995.

[22] V. Van Reijswoud. *The Structure of Business Communication: Theory, Model and Application*. PhD thesis, Delft University, 1996.

[23] E. Verharen. *A Language-Action Perspective on the Design of Cooperative Information Agents*. PhD thesis, Infolab, Tilburg University, 1997.

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

[25] S.K. White. *The Recent Work of Jürgen Habermas: Reason, Justice, and Modernity*. Cambridge University Press, 1988.

[26] T. Winograd. 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.

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

[28] V. Wulf and M. Rohde. Towards an integrated organization and technology development. In *Proceedings of the Symposium on Designing Interactive Systems, Ann Arbor, USA, August 23-25, 1995*, pages 55–64. ACM Press, New York, 1995.