

# A Framework for the Normative Analysis of Workflow Loops

Hans Weigand and Aldo de Moor

Infolab, Tilburg University, The Netherlands  
{weigand|ademoor}@kub.nl

## Abstract

The goal of this paper is to make the communication norms underlying various LAP workflow loop models (DEMO, ActionWorkflow) explicit and to contrast them with auditing norms. We conclude that the OER-paradigm embedded in DEMO and the customer satisfaction orientation of Action Workflow lead to norms which resemble the ones required by internal control, but there are some important differences. We propose a framework for the normative analysis of workflow loops in which customer relations and agency relations are distinguished. Whereas most LAP approaches do not take agency relations explicitly into account, the extended workflow loop model allows us to analyze the effects of delegation on communicative structures.

## 1 Introduction

ActionWorkflow and DEMO are two approaches that offer a special modelling method for business processes based on the Language/Action Perspective. In contrast to data-oriented methods such as state transition diagrams, or UML cases,

---

*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 Sixth International Workshop on the Language-Action Perspective on Communication Modelling (LAP 2001)**

Montreal, Canada, July 21-22, 2001  
(M. Schoop, J. Taylor, eds.)

<http://www-i5.informatik.rwth-aachen.de/conf/lap2001/>

the modelling is based on the notion of a speech act. Moreover, the modelling method imposes a certain structure on the communication processes. In the case of DEMO, this is the transaction paradigm, in the case of ActionWorkflow, the ActionWorkflow loop. This imposed structure excludes certain “ill-formed” processes. Data-oriented approaches do not impose much: it is not difficult to draw a use case that is syntactically correct, but does not make any sense as communication. Some process-oriented approaches in business process modelling are based on Petri Nets. Petri Nets have the advantage that formal verification techniques can be used to test certain properties. However, a Petri Net in itself does not impose more communication structure to the process than a data-oriented approach.

A major advantage of the LAP approaches – the structure they impose – is sometimes also a point of criticism. According to some researchers, the workflow loop is too restrictive [Suchman, 1994]. It is said that in practice the analyst is confronted with situations that do not adhere to the workflow loop principle. The crucial question is not whether such situations occur, but whether such a deviation is ok or bad. If the deviation is OK, then apparently, LAP is too restrictive. If the deviation turns out to be a distorted communication process, then it is an advantage that the LAP model indicates how this process must be redesigned. However, in order to make a strong case for the advantage of such a prescriptive application of the model, it is essential that the normative principles underlying it are explicated.

The objective of this paper is to explicate the norms inherent in the LAP models, in particular, DEMO and ActionWorkflow. Section 2 introduces the notion of norm-based analysis based on Stamper’s semiotic approach. Section 3 provides a brief overview of the mentioned LAP models. In section 4, an overview is given of communication norms derived from the internal control theory used in accountancy. In Section 5, we make the norms underlying LAP workflow models explicit and compare them with internal control norms. Section 6 introduces our framework for the normative analysis of the workflow loop paradigm, combining elements from the approaches discussed.

## **2. The role of norms in workflow modelling**

Today’s Internet-age information systems are much more communication than computation systems. There are many applications that support complex communication processes, like discussion and group decision making, and many kinds of collaborative work such as group authoring. The semiotics of these systems are often much more complex than of traditional information systems, particularly because the intended semantics and pragmatics are not under the control of one single organization, and therefore often remain un(der)defined. This

entails that often the meaning of information produced and responsibilities for system use and specification are not clear.

In order to deal with such problems, we need to move away from the traditional information *flow* paradigm, in which positivistic modelling aimed at producing automated solutions is central. Instead, an *information field* paradigm is needed [Stamper, 2000]. At the core of this semiotic paradigm are *fields of norms*, binding together groups of people. The norms allow meaning and responsibilities to be clearly specified, thus fostering the active construction of social reality, shared understanding and mutual commitments. Let us explain this using Habermas' theory of communicative action [Habermas, 1984] and the Language/Action Approach presented in [Winograd & Flores, 1986]. For Habermas, norms are grounded in the social domain, and these norms play a role in determining the appropriateness of communicative actions, as opposed to the truth or sincerity. However, this is not the only place where norms play a role. Central in Habermas' framework are the validity claims raised by a communicative action. The truth claim of a statement is not self-evident, but can be discussed. This presupposes that the communicative partners agree (or can arrive at agreement) on certain rules: rules for valid inferences (e.g. modus ponens), rules for interpretation, rules for perception etc. These rules are also called norms in the semiotic model. At first glance, this might sound a bit extreme. This is because in normal (sic!) circumstances, we take these rules for granted and we apply them automatically. However, when a *breakdown* occurs – for example, two communicating parties disagree on the meaning of a term, or on the classification of the product – it becomes clear that the rules are not self-evident, and they are revealed as norms. Since they are norms, they can be violated (even if that would be exceptional) and the communicating parties can be held responsible. Note that “norm” should not be interpreted in the narrow sense of laws or ethical rules imposed by some society or institution. A norm is any rule that we apply in our daily practice and that we expect others to apply.

In [De Moor and Weigand, 2001], semiotic norms are used to ground a model that can be used to assess and improve the quality of business communication processes. Instead of focusing on the meta-processes involved in improving the quality, we are interested in this paper in grinding a diagnostic lens specifically designed for detecting workflow loops complexities and deficiencies. A communications quality model grounded in organizational semiotics could then use such a normative basis as an input.

To find the sources of norms that can guide communicative workflow action, we start with a normative analysis of LAP-based workflow modelling methods. A normative analysis is aimed at making underlying norms explicit. This cannot be done without interpretation. We will do this to the best of our knowledge, but it goes without saying that the responsibility for this interpretation is our own.

### 3. Business modelling – DEMO and ActionWorkflow

To build a normative framework for analyzing workflow loops, we analyze two well-known LAP-business modelling approaches: ActionWorkflow and DEMO (based on the OER paradigm).

#### 3.1. ActionWorkflow: the customer orientation

ActionWorkflow ([Medina-Mora et al., 1993; Denning & Medina-Mora, 1995]) is a theory about the organization of work taking a LAP and relies on theoretical work of [Winograd and Flores, 1986]. ActionWorkflow can be seen as generic business framework, or a business process and workflow analysis and modelling method, and is also the name of a supporting software tool. It uses the ‘work is a closed loop’ idea (Figure 1). According to [Denning & Medina-Mora, 1995], traditional workflow management methods have been *production*-centered, focusing on efficiency and control, whereas their approach is *satisfaction*-oriented, with a central focus on commitments, conditions of satisfaction and timely completion.

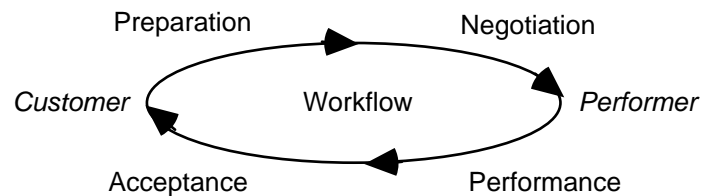


Figure 1: ActionWorkflow

Business processes are split up in elementary transactions between a customer and performer and consist of the steps: preparation, negotiation, performance, and acceptance. The first two steps aim at the establishment of a commitment of the addressee to perform an action. The last two steps aim at the establishment of the performed action. The action itself is not modelled, only its results. In both parts there is negotiation aimed at mutual agreement of what has to be established. The ActionWorkflow theory (with its roles and phases) can be seen as a generic blueprint for the organization of work.

The ActionWorkflow is recommended for improving the customer satisfaction; it is realized that in practice, the loop is often not closed. “Incomplete work flows invariably cause breakdowns, and if they persist, they give rise to complaints and bad feelings that interfere with the ultimate purpose of work – to satisfy the customer” [Denning & Medina Mora, 1995]. It is stated that “many of the

problems that plague organizations are connected with persistently incomplete work flows”. Indeed, incompleteness of workflows is an important although not the only possible deficiency.

For the purpose of norm analysis, we can conclude that “closing the loop” is a very important underlying norm of the Action Workflow approach. The approach incorporates some important principles: (a) work in organizations is done *for or on behalf of* somebody, (b) task assignment should be followed up by task evaluation, and (c) facts derive from actions having been performed, so fact creation (as part of the task evaluation) is preceded by task assignment. These principles will come back in section 5.

### 3.2 DEMO: The OER-paradigm

DEMO (Dynamic Essential Modelling of Organizations) [Dietz, 1994] is a business process modelling method based on social theory, grounded in the philosophy of Searle and Habermas. The motivation behind DEMO is the strongly felt need to have a theory about the dynamics of activities in organizations for IS analysis.

According to Dietz communicative acts in business communication are related to each other according to a specific pattern, called the transaction pattern. The pattern consists of a communication part and an action part (see Figure 2).

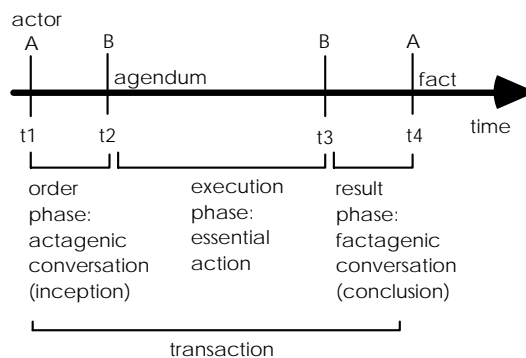


Figure 2: Transaction pattern  
(after [Dietz, 1994])

The transaction starts with a request of the initiator A (at time t1). The participants involved in the transaction, called actors, reach (at t2) a commitment for a future action, called agendum (thing-to-do), added to the agenda for the actor involved. Next, the action agreed upon is executed by the executor (t2-t3). Finally, the parties try to reach an agreement about the result of the action. When the initiator accepts the result, the transaction succeeds and a fact is created (t4). The fact corresponds with the predication of the communication act as mentioned

above. According to Dietz, the essence of the behaviour of an organization consists of the continuous accomplishments of such transactions between actors.

The Action Workflow loop is, seen from a distance, very similar to the OER transaction pattern. However, we must note that there is an important difference in the presentation. Whereas Action Workflow motivates the use of workflow loops by the problems that occur when the loops are not closed, according to DEMO, the transaction pattern just models how things are. If an actagenic conversation is found followed by some execution, but without factagenic conversation, the DEMO approach concludes that there is no transaction in this case, but DEMO makes no statement on whether this is good or bad. On the other hand, the OER paradigm considers it the normal case that communication is realized in transactions. For that reason, the DEMO handbook [Dietz & Van Reijswoud, 1998- p.51] can state that often some of the communicative actions making up a transaction are missing. "But if there is at least one communicative action, the other must be there also, in one way or another". This "must" is not meant as a normative rule, but as a design heuristic. Nevertheless, we can conclude that DEMO considers it to be the normal case that the loop is closed.

So although DEMO itself does not want to attach normative value to the transaction pattern, a norm analysis must conclude that the same underlying principles that we mentioned in the above are applied. However, the norms are weak. The fact that DEMO allows communicative actions to be missing or implicit, shows that apparently, other norms, such as efficiency, can easily overrule the communicative norms.

A feature of the DEMO approach, and particularly the transaction pattern, is that it is highly abstract. Abstraction is nice in a certain phase of the design process, but can go at the expense of concrete guidance on how to improve communication structures. For example, the roles initiator and executor can be fulfilled by one or more subjects. Therefore, it is possible that the initiation (starting the actagenic conversation) is performed by one subject, and the evaluation (closing the factagenic conversation) by another – they are performing together the role of "initiator"- and it is also not excluded that initiator and executor roles are assigned to the same subject. It is not easy to say whether DEMO prefers the initiator and evaluator to be the same subject – what Action Workflow strongly promotes – although the fact that there is only one role suggests that there must be some close connection. But it is not more than a suggestion.

DEMO claims that it can be used in business process reengineering [Dietz, 1994]. Our claim is that business process reengineering assumes that we have some principles that allow us to say that this particular realization is better than another. For that purpose, we try to make explicit the implicit norms in LAP approaches as well as extend them with norms currently not taken into account.

#### **4. Internal control theory**

The information system in an organization provides the information to its actors to execute and coordinate their tasks. According to audit theory, an organization does not only need an information system, but also an internal control system to secure trustworthiness of the registered information and to control potential errors. According to [Starreveld, 1997], internal control is needed when an organization has a delegated task structure which allows agents to establish commitments on behalf of the organization, to employ certain funds, goods or products. The principal that has delegated such activities will have the evident need to control the agent that performs these activities. However, delegating an activity does not mean that the responsibility for this activity is delegated as well. Instead, it introduces a control task for the person that delegated a task to another person. This is where, among other reasons, communication comes in: since the principal responsible for that control task cannot personally observe the performance of operating tasks, he must rely on documentary evidence (evidence function). On the other hand, to protect himself, the executing party (the agent) must be able to prove the completion of an activity (preventative function).

Chen ([Chen,1992]) has defined a couple of control principles based on his review of the accountancy literature that are worth citing:

1. If an agent-based operational task exists, its corresponding control task should exist as well and should follow the operational task.
2. If a control task exists, it must be furnished by supporting documents. These supporting documents should be the result of a previous control task that directly witnesses the activity to be controlled.
3. Supporting documents should be generated by a source independent of the source which generates the document to be verified.
4. If a control task uses a supporting document, this should be transferred directly from the control task which verified it.
5. An operational task and its corresponding control task should be segregated into two different organizational positions and into two different agents.
6. An operational task and its corresponding control task must not be lower in the formal power hierarchy than the position responsible for the operational task.
7. The agents responsible for the operational task and its corresponding control task should be socially detached.

[Bons,1997] notes that control tasks can be divided into two categories: control tasks that make direct statements about the operational tasks, and control tasks that evaluate the resulting document and draw conclusions based on them.

The principles of Chen focus on control tasks. Starreveld (ibid) states the fundamental principle that agents should render account of the tasks delegated to them. This is both an imperative from the side of the principal – since the principal delegated certain authorizations and access to resources to the agent – and from the side of the agent there is a need to receive discharge from his responsibility. This does not take away the need for evaluative control – to the contrary, this control task builds on the account of the agent himself and completes it.

Some of the above principles are only relevant in an intra-organizational context (in particular number 6), but there are also some principles that become especially relevant in an inter-organizational context. In particular, concerning the contracting process that results in certain obligations for the operating party. [Dewitz,1992] has defined the following rules for that part:

1. The role issuing the promise should be responsible for the primary activity being promised
2. The beneficiary of this activity should receive the document (with the promise).

Finally, Bons formulates his own general principles of inter-organizational controls based on the ones of Chen and Dewitz, but explicitly pays attention to the implicit or explicit outsourcing of activities, and to the reciprocal character of a business contract. Such a contract requires some rules on the performance of one's own activity in relationship to the counteractivity. In this paper, we will limit ourselves to the asymmetric case such as represented in the LAP approaches.

## **5. Making the workflow loop paradigm norms explicit**

We are now in a position in which we can analyze the norms inherent in the OER-paradigm and ActionWorkflow approach (together called workflow loop paradigm in the following) and relate them to principles of internal control. In ActionWorkflow, the initiator is called *customer*. The differences between an agency relation and a customer relation will become clear during the following discussion. However, to compare the two we must assume, for the time being, that the norms are the same in both situations.

In 5.2 we shall come back on the problems that occur when combining a customer loop and an agency loop. Next, we first make explicit the implicit norms embedded in the workflow loop paradigm.



### **5.1 The workflow loop norms**

In our view, the workflow paradigm incorporates at least the following normative principles (we follow Chen's list and already use some terminology of internal control theory in the phrasing of these principles):

1. For any activity, a distinction must be made between the *operational task* and the *control task*. These two tasks are executed by two different roles and two different subjects.
2. If an operational task exists, there should be a corresponding *initiating task* and the operational task should follow the initiating task.
3. If an operational task exists, its corresponding control task should exist as well and should always follow the operational task.
4. The initiating task should contain a request for action from a role (initiator) independent of the role performing the task
5. The role issuing the initiating task (initiator) should be the same as the role responsible for the (evaluative) control task.
6. The initiating task should be closed with a commitment (promise) from the role performing the operational task.
7. The control task should be furnished by supporting documents. The supporting documents should originate from the role performing the operational task.
8. The control task should be closed with a performative statement from the role performing the control task.
9. The performative statement of the control task should be received by the role that performed the task.

Ad 1. The first principle corresponds to principles 1 and 5 of Chen. In OER even more clearly than in ActionWorkflow, the transaction consists of an execution part and a control part. That the initiator and executor are two different roles and subjects, is not stated explicitly, but is implicit in the transaction design. In this way, evidence is produced for the initiator that the action has been performed, and the executor is discharged from the obligation (and can prove that later).

Ad 2. The second principle is related to the additional principles of [Dewitz,1992]. The principle can be defended since it would not be desirable that an agent performs a certain task without being requested to do so. Also, for the agent himself it is beneficial, if not necessary, that he is backed up by an explicit instruction when he is later asked why he did perform this particular action. Think in particular of an agent who is made responsible for the access to some goods, such as in inventory manager or a treasurer, and the action consists in giving away

these goods. The main reason that Chen omits this principle is probably his exclusive focus on control tasks.

Ad 3. This principle corresponds to principle 1 of Chen.

Ad 4. This principle is a refinement of principle 2. Although the term “initiator” suggests that this role takes the initiative, we have formulated the principle a bit weaker: the task should contain a request, but this may have been prompted by an offer of the executor.

Ad 5. This principle is central to the satisfaction-orientation in ActionWorkflow, but less obvious from a control point of view. From the control point of view, it is sufficient that the task is (independently) evaluated by the organization, but it does not matter who performs the evaluation. For the customer-orientation of ActionWorkflow it *does* matter; if the evaluation is performed by a third party, or the boss, this may be sufficient for preventing fraud and for discharging the agent, but it may not be sufficient for the customer being satisfied.

Ad 6. This principle corresponds to the principles of Dewitz. For the principal/beneficiary, a commitment is important, and obviously the commitment should be given by the agent that performs the action.

Ad 7. This principle seems to contradict principle 3 of Chen that not the agent himself, but another party should provide evidence of completion. However, the contradiction can be resolved by observing that the account provided by the agent himself is not taken at face value, but is evaluated and verified in a next step. In this final evaluation step, the initiator can use third-party evidence. The organization should have certain norms that mediate between the execution of the material act and the verbal report of that act. These norms should foster objectivity and prevent fraud.

Ad 8. This principle has some relation to principle 2 of Chen, but is stronger. It occurs to us that it is important for discharging the executor and hence the fact that it does not occur in Chen’s list, must be an omission on his part.

Ad 9. This principle is open for discussion. It is important that the executor is discharged, but it could be sufficient if the performative document (the evidence) is passed to some independent party. The practical advantage of the principle is that the executor can now consider the action as closed, instead of keeping the file open.

Summarizing, we can conclude that there is a big overlap between the normative principles underlying the workflow loop paradigm and the internal control principles of Chen. However, some internal control principles are missing, such as the ones related to conflicts of interests, while on the other hand, some principles of workflow loop model are stronger, due to its customer-orientation. Thus, an analysis of the workflow loop models from an internal control perspective is clearly useful in getting a clearer view on how to use them in a normative, prescriptive way. Still, a simple workflow loop angle is not sufficient to deal with some organizational complexities found in business and networked enterprises, since a mere focus on customer-performer interaction is not sufficient. An example is the communicational complexities caused by the delegation of agency. An analysis of this problem will be conducted in Section 6.2.

## **6. A framework for the normative analysis of workflow loops**

Based on the previous discussion, we now create a framework that can be used to analyze the normative content of workflow loop models. We first outline the framework, then apply it to an analysis of the agency problem sketched in the previous section.

### **6.1 Outlining the framework**

The framework takes the contractual relationship between two actors as its starting point. In this contractual or reciprocal relationship, there are at least two services exchanged. The performer of one service is the beneficiary of the other. The next step is that one actor delegates his task to an agent. In this paper, we will focus on delegation of the service execution, but delegation at the beneficiary's site is possible as well. Although the task, or part of it, can be delegated to an agent, the delegating actor still keeps a relationship and responsibility to the other party<sup>1</sup>. We call this a *contractual* relationship, whether there is a written contract or not. ActionWorkflow provides us with the *functional roles* of customer and performer. We rename performer into agent and the customer into beneficiary in order to integrate this perspective into agency theory.

From DEMO, we take the *workflow control roles* of initiator and executor, to which we add the evaluator role. Instead of using the specific DEMO (order, execution, result) and ActionWorkflow (preparation, negotiation, performance, and acceptance) workflow loop phases, we use the neutral terminology of

---

<sup>1</sup> An important principle of subcontracting is laid down in Article 8.107 of the "THE PRINCIPLES OF EUROPEAN CONTRACT LAW – 1998" stating that *a party who entrusts performance of the contract to another person remains responsible for performance*

initiation (I), execution (X), and evaluation (E). These are tasks. These tasks are interconnected by conversations: the actagenic conversation and the factagenic conversation. Tasks and conversations together are constitutive of the communication loop. The workflow loop is a specific kind of communication loop.

From internal control theory, we derive the distinction in operational and control tasks. We define the functional role of principal to be responsible for the control tasks. The control loop is also a specific kind of communication loop.

The extended workflow loop model is presented in Fig.3. Note that the agent has two executor roles, but there is a slight difference between the X-role of the agent in the work loop and the X-role of the agent in the control loop. From a control perspective, the agents' performance consists of his executing the wishes of the beneficiary plus his conversations with the beneficiary (so his overall performance on the work loop).

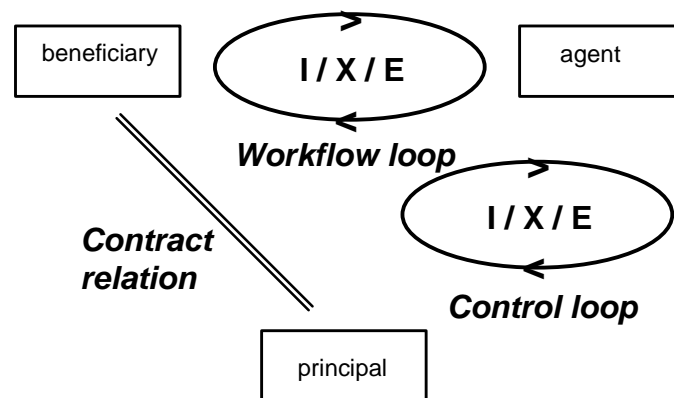


Figure 3: The extended workflow loop model

## 6.2 Motivating the extended model: agency analysis

Agency means that a relation between an agent and some principal exist, where the agent executes work on behalf of a principal. According to [Taylor,1993], an agent has a responsibility for the conduct of an operation. As an agent, he also acts for somebody: the beneficiary. So he has a double responsibility: he acts *on behalf of* someone and he acts *for* (to the benefit of) someone.

According to Taylor, every organizational event is, communicatively speaking, doubly embedded – in the universe of the agent, with his or her preoccupation with instrumentality on the one hand, and, on the other, in the universe of experience of the beneficiary for whom the patient (the good or service

offered) is not just a body to be regulated and operated on, but something affecting his or her well-being.

In practice, we will often encounter the following interpretation of figure 3. There is an employee (agent) who performs some service to a customer (beneficiary) on behalf of an organization represented by a manager (principal). The beneficiary can be inside or outside the organization. In some special cases, like a secretary performing jobs for the boss who hired him, the beneficiary and manager/principal are the same subject.

By focusing on customer satisfaction, the ActionWorkflow approach tends to ignore the agency relationship between manager and employee and the accompanying communication needs. Of course, customer satisfaction can be more important than production orientation and increase of efficiency. Usually, however, both will be important to some extent. The conclusion must be that the workflow loop is useful but must be supplemented with other models that focus on the production optimization (the information flow between principal and agent, consisting of setting production norms, task descriptions as well as accounts, progress reports). Note that in all cases where optimizing a certain value is important, a feedback loop is fundamental – whether this value is customer satisfaction, efficiency, quality or whatever. However, the feedback loop must be specialized for each such objective.

### **6.3 The extended LAP workflow loop norms**

The framework urges us to generalize and extend the previous LAP workflow loop norms. Without further comment we list them here, and illustrate them with a simple example.

1. Any action performed by some actor must have a beneficiary.
2. For any delegation relationship, a distinction must be made between the *operational (work)* and the *control* task. These two tasks are executed by two different subjects, the agent and principal, respectively.
3. If a work task exists, there should also be a corresponding *initiating conversation*. The task should follow the initiating conversation. If the task is a delegated one, the initiating conversation must be backed by a *contract* or *delegation* relationship between principal and initiator.
4. If a work task exists, its corresponding *evaluation* conversation should exist as well and should always follow the task. If the task is a delegated one, the evaluation conversation must be backed by a *contract* or *delegation* relationship between principal and evaluator.
5. The beneficiary should be involved in the initiating conversation and the evaluation conversation.
6. The principal should be both the initiator and the evaluator of the control loop.

7. The control task should be furnished by supporting documents. Supporting documents should be generated by a source independent of the agent responsible for the work task.
8. The control task should be closed with a performative statement from the principal.
9. The performative statement of the control task should be received by or be accessible to the agent.

*Example: a pizza delivery case*

Suppose that a pizza baker (as a performer) originally bakes and delivers his pizzas himself. In that case, the communication between him and a customer can be easily modelled using a standard workflow loop (within a contract relation, but we will focus here on the baker as performer). Now the baker hires a boy to deliver to the pizza to the house of the hungry client for him. Then there exists an agency relation between the baker and the boy: the baker plays the manager/principal role, the boy the employee/agent role. The work loop now seems distorted, since the new pizza delivery work flow loop performer is no longer one subject. Say the hungry client calls the baker on the phone. In an actagenic conversation, part of the workflow loop, the baker agrees to bake and deliver a pizza. After hiring the boy, the baker orders the boy to take the pizza to the client. The boy takes the pizza, drives to the house, rings and starts a factagenic conversation in which the hungry client accepts the pizza, perhaps after having signed a note. The original workflow loop is violated, at least in the sense that the actagenic conversation and the factagenic conversation do not have identical executors anymore (cf. 5.1, principle 6). It might be possible to consider the telephone conversation between the client and the baker as focused on the ordering of the pizza, and as consisting of an actagenic conversation and a factagenic conversation (in which the baker merely states that the order has been placed). This is the way [Steuten,1998] analyzes a hotel room booking communication. However, it is clear that a commitment from the baker is sufficient to finish the call (we can view the affirmation of this commitment as a expression of the fact that the commitment has been created, but this is redundant); moreover, the relationship with the conversation later between pizza boy and client would get obscured.

From a DEMO perspective, the workflow loop is not really distorted, only the subject assignment is more complicated. This is confirmed when we check the norms that we specified. The norms do not require that the workflow loop is delegated as a whole. However, our norms do state that a contractual relationship between baker and customer remains, and that the delegation introduces a control loop. These norms are not expressed in DEMO (and also cannot be expressed at the level of Communication Structures).

## **6.4 Delegation**

The extended workflow model makes delegation processes explicit. To analyze delegation and its effects in more detail, we now first have look at what exactly can be delegated.

Let us start at the level of the workflow loop. In this loop, we identified three tasks and two conversations. The tasks are the tasks of initiating, executing and evaluating. The conversations are actagenic and factagenic. Each conversation consists basically of two communicative actions.

In the example of the pizza baker, the baker delegated (a) part of the executing task – the delivery, and (b) the initiation of the factagenic conversation. In addition, he could delegate the ordering as well – for example, when his wife takes up the phone. In that case, he has also delegated the actagenic conversation. Note that, unless specified otherwise, in both cases he has delegated the success layer part only. When the customer has complaints (that is, when the discussion layer is entered), he goes to the baker, since the contractual relationship is with the baker. Although the baker could also delegate the complaint handling, there will always remain a residue of responsibility since he cannot delegate the contract relationship itself.

The control task of the baker includes initiating and evaluating tasks. The baker can delegate these tasks as well. Or he can delegate the evaluation conversation (the collection of evidential documents) and keep the evaluation task (the decision to decharge the boy's responsibility).

### *Example: the pizza story continued*

Another layer of agency complexity is introduced when the baker not only delegates the delivery of the pizza but also the baking itself, let us say, to his daughter. After having received the phone call, he instructs his daughter to bake the pizza. Then, after the baking, the daughter may report back to the pizza baker. That would create a customer/performer workflow between baker and daughter. However, they both find it more efficient that she directly instructs the pizza boy to deliver the pizza. There are no workflow conversations between baker and boy in this situation anymore, only the control loops (delegation links) remain. And several workflow loops seem to be only partial (only actagenic or factagenic). We will come back on the conversation between daughter and boy shortly.

What about the norms? The first norm states that every action has a beneficiary. But who is the beneficiary of the daughter's baking the cake? It must be the customer, although there is no direct conversation between the two. But this is not strictly required. What is required is that there is some evaluation of the action and that the beneficiary is involved. This would imply that the factagenic conversation between boy and hungry client is an evaluation of the whole workflow process, including the baking.

What about the conversation between daughter and the boy? Let us assume that this takes the form of a simple conversation:

Daughter: Pizza is ready! You can go.  
 Boy: OK

This conversation has in fact two functions. On the one hand, it instructs the boy to deliver the pizza. The boy's OK is then the expression of his commitment to do it. On the other hand, the daughter reports that the baking is finished. Perhaps the hungry client will not be satisfied in the end, but it is a fact that a pizza is born, and this fact is established when the boy agrees with an OK.

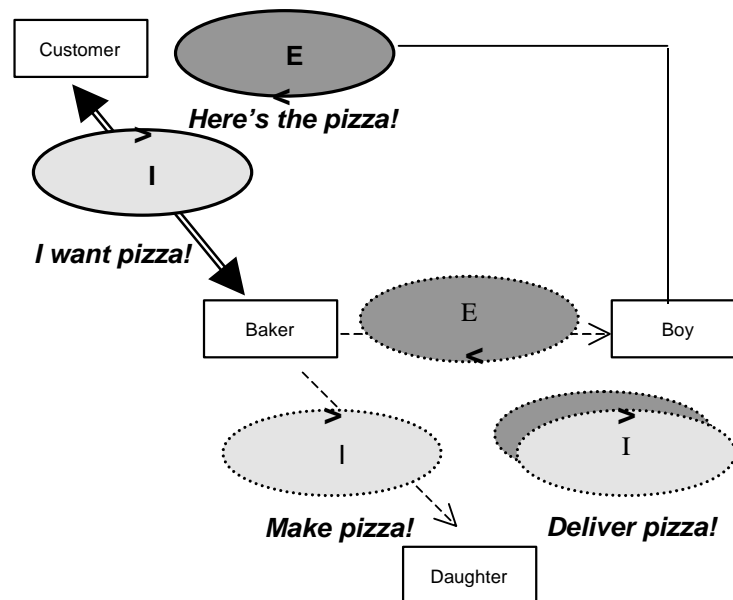


Figure 4: Complex agency in the pizza-delivery case. The delegation lines are indicated as dashed lines. The contractual relationship is indicated as a two-arrow double line. The I or E inside a workflow loop means that there is only one conversation. Control loops are rendered with dotted lines.

So the boy does some evaluation, but we are reluctant to call him the beneficiary. We prefer to interpret this evaluation as a control loop evaluation. The baker, being the principal, delegates part of the evaluation of his agent (the daughter) to somebody else, namely, the boy. The evaluation has a production perspective rather than a customer satisfaction perspective. Note that the conversation between daughter and boy, if documented on paper or otherwise, can



be used later to provide evidence in the sense of rule (7) to discharge the daughter. So it is a factagenic conversation, and together with the actagenic conversation between the baker and the daughter, it “closes the loop” of the control process. At the same time, it is an actagenic conversation with the boy which together with the factagenic conversation between the baker and the boy (after he has returned and submits the delivery sheets), closes the boy’s control loop. But note that although this looks nice, it violates principle 6 that the principal himself should initiate and close the control loop. At this stage, it is not easy to say whether this principle should be relaxed (to allow delegation of control tasks), or whether the situation is really a violation (which could still be preferred for other reasons). This is the kind of questions for future research that follow from our more detailed analysis of the workflow loop and that, in our opinion, are important questions to ask.

It is worth noting that there are closely related alternative realizations. The baker could have realized the relationship with his daughter as a customer satisfaction loop. In that case, he becomes a beneficiary and should express his satisfaction. The present communication lines would not be sufficient then: either the shortcut between daughter and boy should be removed or the boy should have the delegated responsibility to evaluate the baking of the pizza (if delegation is allowed – cf. principle 5 above). Which realization is better cannot be decided on the basis of the present norms. Additional norms are necessary that take into account the efficiency of the communication and other factors such as employee satisfaction.

## **7. Conclusion**

In this paper, an analysis has been made of the norms underlying LAP workflow loop models. Norms implicit in those models have been made explicit and contrasted with explicit norms from internal control used in accountancy. A framework for the (meta)analysis of workflow loop models was created. The framework consists of an extended workflow loop model and a set of reconstructed LAP workflow loop norms.

We claim that an extended workflow model that considers both customer relations and agency relations is needed to chart complex organizational communication situations. LAP, internal control and possibly other norms can be applied to assess the situation. Quality management activities as described in [De Moor & Weigand, 2001] may be used to improve upon the current communication situation. Thus, this framework may prove to be a helpful tool in optimizing organizational communication patterns and semiotics. At least, it allows to formulate important questions that are unexpressible in most of the current business modelling approaches.

There are many things that are still to be done. We do not have a practical way of modelling (diagram technique). Even more important is the way of

working, the way that the model is built up. In our view, there should be a *recursive* method based on the principles of task decomposition, delegation (introducing new agency relations) and outsourcing (introducing new customer relations). For a certain organization or department, one can start with identifying the contract relations with all stakeholders in which the organization is the performing actor. Then the model can be worked out by applying delegation and outsourcing transformations. Each transformation should preserve the validity of the communicative norms, or at least a warning should be given when some norm is violated for some reason. In this way, only meaningful and valid communication structures can be derived. Reengineering can start from a given situation and create a new situation by retracting existing relations and introducing new ones. In both cases, additional quality assessment is needed to determine whether the new situation improves on the old one.

## References

- [Bons, 1997] Bons, R. *Trustworthy trade procedures*. Ph.D. thesis, Erasmus University, Rotterdam, 1997.
- [Chen, 1992] Chen, K.T. *Schematic Evaluation of internal accounting control systems*. Ph.D. thesis, Univ of Texas at Austin, 1992.
- [De Moor & Weigand, 2001] Moor, A. de, Weigand, H. "Towards a Semiotic Communications Quality Model". Proc. IFIP WG8.1 Conf on Organizational Semiotics, 2001.
- [Denning & Medina-Mora, 1995] Denning, P., Medina-Mora, R. Completing the Loops. *Interfaces* 25:3, pp.42-57, 1995.
- [Dewitz, 1992] Dewitz, S.K. *Contracting on a performative network: using information technology as a legal intermediary*. Ph.D. thesis, Univ of Texas at Austin, 1992.
- [Dietz, 1994] Dietz, J. Business Modeling for Business Redesign. Proc. 27<sup>th</sup> HICSS Conference, IEEE Society, 1994.
- [Dietz & Van Reijswoud, 1998] Dietz, J, V. van Reijswoud. *DEMO Modelling Handbook*, version 2., 1998.
- [Habermas, 1984] Habermas, J. *The theory of communicative action*, I. Beacon Press. 1984.
- [Medina Mora et al, 1993] Medina-Mora R., T. Winograd, R. Flores, F. Flores "The ActionWorkflow Approach to Workflow Management Technology". In: *The Information Society*, vol. 9, pp.391-404, 1993.

*A Framework for the Normative Analysis of Workflow Loops*

- [Starreveld, 1997] Starreveld, R.W. *Bestuurlijke Informatieverzorging. Deel 1, Algemene Grondslagen*. Samson, Alphen a/d Rijn, 1997. (in Dutch).
- [Stamper, 2000] Stamper, R. "New Directions for Systems Analysis and Design". In Filipe, J. (ed.), *Enterprise Information Systems*, Kluwer Academic Publ., London, pp.14-39, 2000.
- [Steuten, 1998] Steuten, A. *A Contribution to the Linguistic Analysis of Business Conversations within the Language/Action Perspective*. Ph. D. thesis, Delft University of Technology, 1998.
- [Suchman, 1994] Suchman, L., "Do Categories have Politics? The Language/Action Perspective Reconsidered". In *Computer Supported Cooperative Work (CSCW)*, vol. 2, 1994, pp.177-190.
- [Taylor, 1993] Taylor J.R. *Rethinking the Theory of Organizational Communication*. New Jersey: Ablex, 1993.
- [Winograd & Flores, 1986] Winograd, T. & Flores, F. *Understanding Computers and Cognition: A New Foundation for Design*. Ablex Publishing, 1986.