This paper is published as Chapter 7 in *Information Modeling in the New Millennium,* (Eds, Rossi M and Siau K) Hershey, PA: Idea Group Publishing, 2001, pp. 110-136.

Business Action and Information Modelling
— The Task of the New Millennium

Pär J. Ågerfalk[1,3] and Göran Goldkuhl[2,3]

[1]Dept. of Informatics (ESA)
Örebro University, SE-701 82 Örebro, Sweden
pak@esa.oru.se

[2]Dept. of Computer and Information Science
Linköping University, SE-581 83 Linköping, Sweden
ggo@ida.liu.se

[3]Centre for Studies of Humans, Technology and Organization (CMTO)
Linköping University, SE-581 83 Linköping, Sweden

Abstract
In this paper, we discuss the importance of considering the action character of information when modelling information in business processes. The Language Action Perspective (LAP) is described and proposed as the perspective of choice for information modelling – now and for the future. It is emphasized that two of the most important research areas in the new millennium are to further develop LAP, and to operationalize it into systems development methods. Furthermore, the generic business framework of Business Action Theory (BAT) and the requirements engineering method VIBA/SIMM (Versatile Information and Business Analysis according to the Situation adaptable work and Information Modelling Method), as representatives of LAP, are both described and positioned within LAP. This positioning is achieved by elaborating different LAP approaches and their relationships to BAT and VIBA/SIMM.

INTRODUCTION

The Language Action Perspective (LAP) presents an alternative foundation for understanding information and information systems. Interest in LAP was born in the early 1980s (Flores & Ludlow, 1980; Goldkuhl & Lyytinen, 1982) and LAP gained worldwide popularity during the late 1980s. This was due largely to Winograd & Flores' (1986) book, which proposed LAP as a 'new foundation for design'. Today, at the dawn of a new millennium, LAP has reached a mature level. Several business-modelling frameworks are based on LAP, such as Action Workflow (Denning & Medina-Mora, 1995), SAMPO (Speech-act-based Office Modeling Approach) (Auramäki *et al.*, 1988), DEMO (Dynamic Essential Modelling of Organisations) (Dietz, 1994), and BAT (Business Action Theory) (Goldkuhl, 1998). There are also systems development and requirements engineering approaches, such as Commodious (Holm, 1996), SOMA (Semantic Object Modelling Approach) (Graham, 1998), and VIBA/SIMM (Versatile Information and Business Analysis according to the Situation adaptable work and Information Modelling Method) (Goldkuhl & Ågerfalk, 1998).

LAP is based on the theory of speech acts introduced by Austin (1962), and further developed and formalized by Searle (1969; 1979). Over the years, LAP has also incorporated other sociolinguistic theories, of which the theory of communicative action by Habermas (1984) is the most prominent. The main message of LAP is that an utterance (a communication action) carries more than mere facts that tell something about something. Utterances also carry the speaker's intentions and beliefs. We **do** things by speaking. Austin (1962) coined the phrase 'descriptive fallacy' referring to the misconception that language is used only for descriptive purposes. He argued that language is also used for other purposes, such as promising, commanding, requesting, and so on. Thus, according to LAP, a message communicated from a speaker to a hearer consists of two components: a *propositional content* and an *illocutionary force*. The propositional content represents what is talked about, and the illocutionary force implies the type of relationship established between speaker and hearer.

When using an information technology (IT) based information system (IS) in a business, communicative acts are performed through the system. Therefore, information systems are not considered 'containers of facts' or 'tools for information transfer'. Instead, information systems are considered 'vehicles for communication' among people and organizations (Ågerfalk & Goldkuhl, 1998). In order to perceive information according to LAP and to build information systems that support this communicative view of information and business, we need to develop frameworks and method support that explicitly take the 'communicative dimension' into consideration. As mentioned above, there are several mature frameworks and methods available today that aim towards these views. To date, however, most LAP research has focused on generic communication modelling, with business modelling as the primary application. Consequently, the operationalization of LAP into systems development methods is one of the key tasks for the future.

One recent approach that aims to make explicit use of accumulated LAP knowledge for integrating business modelling and systems development is VIBA/SIMM (Goldkuhl & Ågerfalk, 1998; Ågerfalk & Goldkuhl, 1998; Cronholm *et al.*, 1999; Ågerfalk *et al.*, 1999; Ågerfalk, 1999). VIBA/SIMM has been designed to fit within BAT (Goldkuhl, 1998), and is founded on the concept of 'actability', which can be thought of as an action-oriented enhancement of usability.

Within this framework, a business process is regarded as an institutionalized pattern of communication actions aimed at creating value for the business' customers. This means that business actors perform actions, linguistic and non-linguistic, to fulfil the business' goals. Such actions are structured in a predefined way to guarantee that the different quality criteria are met and that the business process is effective and efficient. Communicative actions can be performed manually by business actors, or in interaction between business actors and information systems, or they can be performed automatically by information systems according to predefined action patterns. In order to be successful as a tool for communicative business action, an information system shall possess a high degree of *actability*. Actability has been defined by Cronholm *et al.* (1999) as 'an information systems ability to perform actions and to permit, promote and facilitate users to perform their actions both through the system and based on messages from the system, in some business context'. Thus, actability differs from usability in that it explicitly addresses the action character of information and information systems as promoted by LAP (*ibid.*)

In this paper, we will present and elaborate on both BAT and VIBA/SIMM as representatives of LAP which, we believe, is the perspective of choice for the future. Whitaker (1992) has stated that: 'The greatest impact of alternative linguistic models on IT has been that of Austin's theory, as elaborated by Searle and evangelized by Winograd & Flores'. In the same fashion, we wish to add that carrying on the tradition and further developing and applying this theory is one of the most important tasks that information modellers face to be successful in the new millennium.

The paper is organized as follows. Firstly, we will present and discuss LAP and the importance of considering the action character of information when developing information systems of high actability. Secondly, we will discuss different generic business frameworks within LAP with a focus on BAT and integrated development of businesses and business supporting information systems. Finally, we will outline VIBA/SIMM as a language action approach to requirements engineering following BAT, and relate it to the (somewhat similar) Commodious method.

INFORMATION AND INFORMATION SYSTEMS

Information as Action

Many approaches and theories within the field of information systems are based on a strict representational view of information. This is the case, for example, with traditional information systems theory (Langefors, 1966), E/R modelling (Chen, 1975), Structured Analysis (Yourdon, 1989), and various object-oriented methods (e.g., Booch, 1994; Blaha & Premerlani, 1998). From such a viewpoint, one of the main purposes of requirements elicitation and analysis is to obtain an accurate 'image' of the problem domain in order to have that piece of reality properly represented in the future (data base of the) system. However, such a strict representational view seems to be an instance of Austin's (1962) 'descriptive fallacy' and can be challenged in several ways. Admittedly, one obvious application of language use is to describe things, that is, uttering statements about the nature and state of the world. However, language is also used for many other purposes. Language is used to command, request, question, promise, warn, assign, *et cetera* (Austin, 1962; Searle, 1969; Habermas, 1984). A descriptive fallacy is said to occur when language is considered as being used *only* for descriptive purposes, which seems to be the case with the strict representational view of information (cf. Holm (1996) on '*the technological version of the descriptive fallacy*'). In the same way as for language, information systems are used for purposes other than pure description. From this language action point of view, information systems should not be considered as pure 'fact repositories'. Information systems should instead be seen as vehicles for communication among people and organizations.

The main manifesto of LAP is that an utterance (message) is a combination of both a propositional (informational) content and an illocutionary force (an action mode). The propositional content is what is talked about and the illocutionary force refers to the kind of action being performed (Searle, 1969). The illocution used is a result of the intention behind the communication. When we communicate, we formulate a propositional content and embed this in a communicative action type.

A propositional content, as described in Fig. 1, can be used in many different utterances (messages). It can be used in offers, orders, promises, contracts, predictions, questions, reports, and possibly other types of communicative acts.

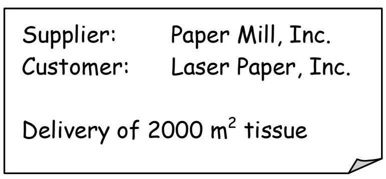


Fig. 1: Example of a propositional content that can be embedded in different types of communicative acts.

Several attempts to classify speech acts with respect to illocution have been performed (e.g., Searle, 1979; Habermas, 1984). Although such classification is not the key point in our adoption of speech act theory, Table 1 shows some examples of how the propositional content of Fig. 1 can be embedded in different types of illocutionary acts (i.e., with different action modes).

In the examples, the same phenomena are referenced, but the meanings of the utterances are obviously quite different.

Action mode	Example utterance
Question	Is it possible for Paper Mill Inc. to deliver 2000 m^2 of tissue to Laser Paper Inc.?
Offer	Paper Mill Inc. offers to deliver 2000 m^2 of tissue to Laser Paper Inc.
Order	Laser Paper Inc. directs Paper Mill Inc. to deliver 2000 m^2 of tissue.
Promise	Paper Mill Inc. will deliver 2000 m^2 of tissue to Laser Paper Inc.
Contract	Laser Paper Inc. and Paper Mill Inc. agree on the delivery of 2000 m^2 of tissue.

Table 1: Examples of different action modes.

Langefors (1966) proposed the concept of 'elementary message' (e-message) to represent the smallest unit that carries information. Any e-message is of one of two possible types. A *property e-message* is a triple (o, p, t), where o is a reference to an object, p is a property of that object, and t is a point in time or a time interval. A *relational e-message* is a triple ((o_1, o_2, …, o_n), r, t), where (o_1, o_2, …, o_n) references n objects, r is a relation between those objects, and t is a point in time or a time interval. The idea is that all information can be decomposed and represented as e-messages of these two types. However, the concept of the e-message is representative of the descriptive notion of information that was criticized in the previous section (cf. Goldkuhl, 1995). Therefore, when talking about communicative action, we prefer to use the concept of 'action elementary message' (ae-message) to refer to the result of a speech act. An ae-message can be regarded as an action-enhanced version of the traditional e-message. However, an ae-message is not an e-message with an augmented illocutionary force, since an ae-message usually references a larger part of the universe of discourse than a single e-message. Whilst an e-message is the smallest unit that carries information, an ae-message is the smallest unit that carries an illocutionary force (action mode), and hence typically contains several e-messages.

Note that in this paper we will use the terms 'ae-message' and 'message' interchangeably.

Information Systems as Action Systems

Information modelling and systems development should be understood in the light of the basic assumption that information systems are used within organizations to support business communication and, hence, the business actions performed. It is important to note, however, that material actions also play a vital role in any business. A material action is an action directed towards some physical object in the business (affecting, producing or consuming). However, in many cases a material action also implies a communicative action. For example, delivering a product, which is a material action, might also be a declaration of the completion of some delivery promise.

From our point of view, information systems have the possibility of performing communicative actions that are governed by specified rules. Those rules can be interpreted as the *action potential* of the system. An IS has an *action repertoire* predefined by such action rules. When performing IS actions, it is often necessary to have access to other earlier actions. For example, when checking an incoming delivery it should be possible to access the corresponding order. An IS must therefore hold a *memory of actions* already performed and other important prerequisites for action.

As a system for communicative action, an IS will, of course, hold a vocabulary of the action topic. Different concepts and their associated terminology are part of the system's action potential. This means that there is a conceptualization of what talk has taken place and what has been talked about.

An IS has a dual action character. It may (1) be an instrument (tool) for users to perform actions, and (2) perform actions independently of its users (although, of course, not independently of its predefined rules). A user (for example, an order clerk) can place an order assisted by an order information system (1). An IS can also perform actions by itself, such as checking for delivery possibilities (2).

In this view, information modellers and system designers perform a special kind of communicative action (regulative action) when defining the discourse for action and communication both through, and within, an IS. That is, they are defining a discourse (an action space) for the possible *interactive* and *automatic* actions, as well as for the possible interpretations of the actions performed. Such interpretations serve as a rationale for subsequent *consequential* actions. Hence, designers are defining the action potential of the system (see Fig. 2).

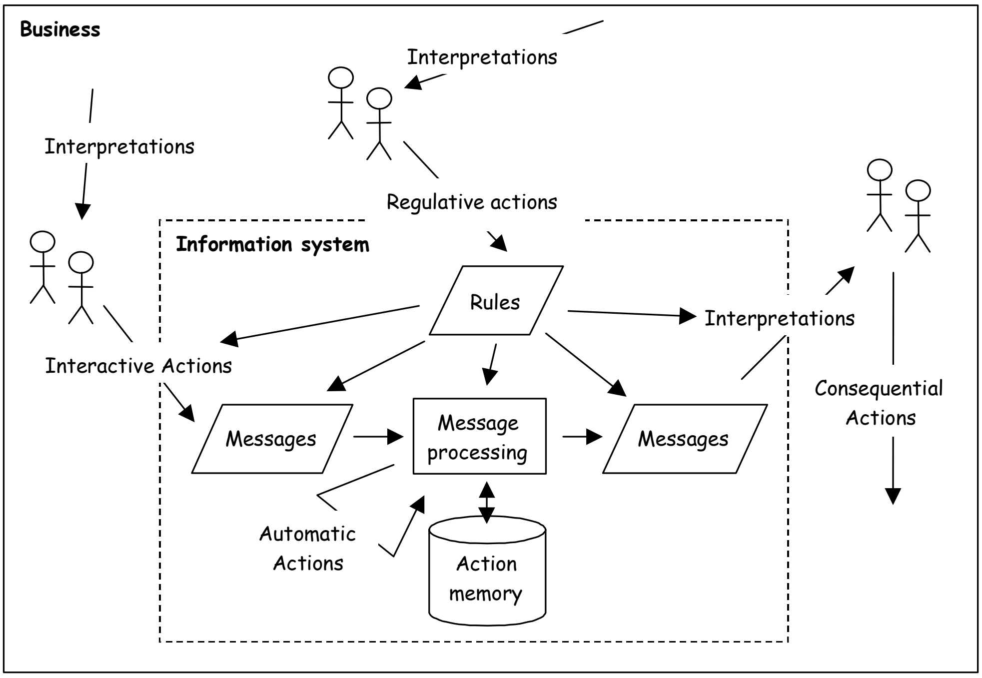


Fig. 2: Information system as action system (Ågerfalk, 1999).

The three types of actions performed when an IS is *used* (interactive, automatic, and consequential), form what we refer to as IS usage situations, which is the topic of the following subsection.

Information System Usage in Business Processes

Within our understanding of IS development, the traditional separation of business modelling and systems modelling (e.g., Jacobson *et al.*, 1995) is inadequate to achieve information systems that are truly integrated with the business. Instead, the design of information systems should be regarded as an integral part of business design (Ågerfalk *et al.*, 1999). The basic assumption is that all system development affects the way business is performed. Hence, by viewing system development as business development, the business effects become conscious and well managed instead of unpredicted and *ad hoc* (Goldkuhl & Ågerfalk, 1998).

As stated in the previous subsection, it is possible to distinguish between three different kinds of IS usage situations within business processes: interactive, automatic, and consequential. The most complex of these three, with respect to information modelling, is the interactive usage situation, which we will therefore focus on in this paper.

Basically, an interactive usage situation consists of formulating and sending ae-messages. It is important to note that any usage situation is part of a larger discourse that constitutes the business process being designed or analysed.

Both the formulation and execution (sending) of messages should be supported by the IS in order for it to be regarded as actable. Ultimately, the IS will help the user to choose what business effects to strive for, choose the action that leads to those effects, formulate the ae-message that corresponds to the chosen action, execute the action, and, give feedback regarding the action performed. Fig. 3 shows how the IS initially presents some information to an interacting user, then the message is formulated in a dialogue between the user and the IS, and finally the message is sent in order to cause some business effect. (Ågerfalk *et al.*, 1999)

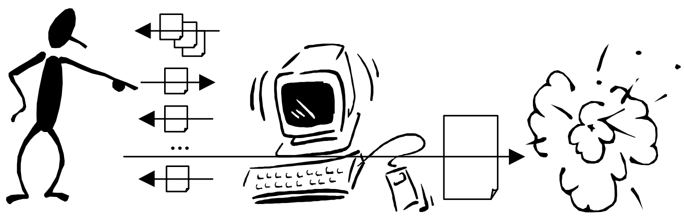


Fig. 3: Human-computer interaction as formulation and sending of messages (Ågerfalk *et al.*, 1999).

Note how the interactions follow a three-phase structure of (1) user action, (2) IS action, and (3) interpretation. We refer to such a structure as an elementary interaction (e-interaction) (cf. Ågerfalk *et al*. (1999) and Ågerfalk (1999)).

BUSINESS COMMUNICATION AND BUSINESS MODELLING

Now that we have established an understanding of information systems as systems for communication, it is time to take one step back and to discuss how business communication and business modelling relate to information and action.

Generic Business Frameworks

Generic business frameworks are general descriptions (conceptions) of business activities and business communication. Such frameworks can be used as templates to direct analysts' attention to certain (important) aspects when modelling a business. Probably the most well-known generic business framework is that of Action Workflow (Action Technologies, 1993; Denning & Medina-Mora, 1995). Two other generic business frameworks are SAMPO (Auramäki *et al.*, 1988) and DEMO (Dietz, 1994). A more thorough model, however, is the more recent BAT model (Goldkuhl, 1998). These four frameworks build explicitly on the language action perspective and hence they help us to emphasize the actions performed by, and within, organizations.

Generic business frameworks differ from 'traditional' modelling techniques such as Data Flow Diagramming (Yourdon, 1989), Use Case Analysis (Jacobson *et al.*, 1995) and Activity Graphing (Lundeberg *et al.*, 1981). The difference is that 'traditional' modelling techniques are what could be called 'notation-driven'. That is, they provide a set of diagramming techniques with associated syntactic and semantic conventions (rules) to describe business processes. Generic business frameworks, on the other hand, are theory-driven. In addition to providing modelling techniques, they also provide a theory of business activity and hence enable a more critical and reflective way of working with business modelling. Usually, generic business frameworks have a more restricted syntax for their notations and also allow different notation techniques to be used.

Action Workflow

The Action Workflow approach (Action Technologies, 1993; Denning & Medina-Mora, 1995) can be used for the description of business interactions between a customer and a supplier. The underlying concept is that whenever a business activity is performed in order to satisfy a customer, a generic pattern of speech acts occurs.

This generic pattern (or schema) of 'conversation for action' has been described by Winograd & Flores (1986) in terms of states and transitions (see Fig. 4), where each transition corresponds to a speech act, and each state corresponds to a state of the conversation.

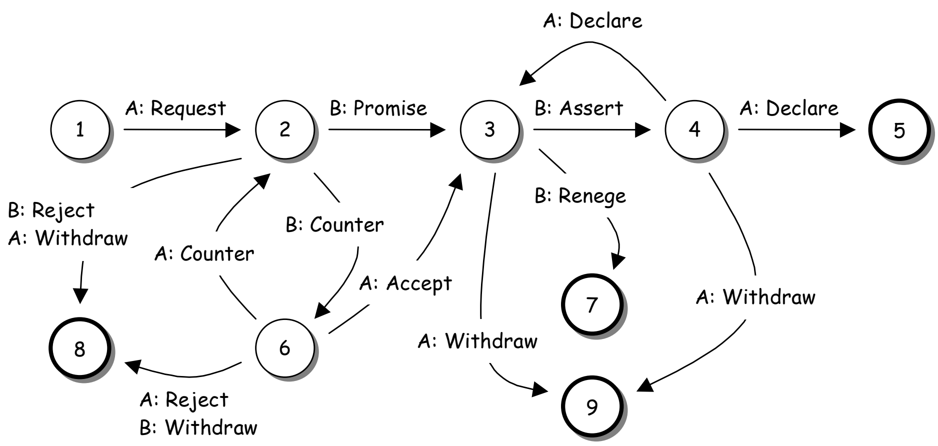


Fig. 4: The basic conversation for action (Winograd & Flores, 1986).

According to the conversation for action schema, a business conversation is initiated by a request from a customer A (the initial speaker), who specifies some conditions of satisfaction. The supplier B (the initial hearer) then has the choice to accept the conditions (promise to satisfy the request), reject it, or make a counter-offer. If, and when, the parties have agreed, then the supplier asserts that the conditions of satisfaction have been met (reporting completion). Now the customer can either declare that, in his or her opinion, the conditions have not been met, or declare satisfaction (which would end the conversation happily). During the conversation, both the supplier and the customer can withdraw at any point and thus cancel the conversation sequence unhappily.

Building on the generic speech act pattern of the conversation for action, the Action Workflow approach describes business interaction as consisting of four phases: (1) preparation, (2) negotiation, (3) performance, and (4) acceptance. As before, the roles are predefined as customer and performer. These phases and roles are described by the so-called Action Workflow loop (see Fig. 5).

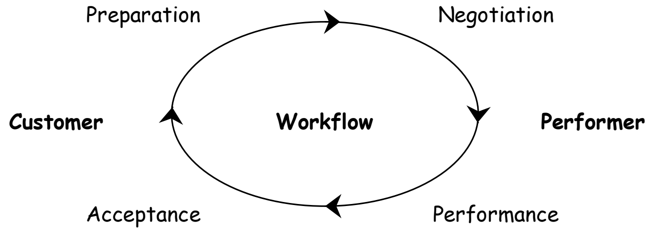


Fig. 5: The Action Workflow loop (Action Technologies, 1993).

The Action Workflow loop can therefore be regarded as a generic conception of the structure of business activity that is used to direct the analysts' attention to the action character of doing business.

DEMO

Another generic business model is DEMO (Dynamic Essential Modelling of Organisations) created by Jan Dietz and his colleagues (e.g., Dietz, 1994; van Reijswoud, 1996). There are obvious similarities between DEMO and Action Workflow. In DEMO, there are two generic roles called *initiator* and *executor*. One main notion is the transaction, with a transaction consisting of three phases: (1) the order phase (where the initiator requests and the executor promises); (2) the execution phase; and (3) the result phase (where the executor states what has been executed and the initiator accepts this). DEMO builds on a distinction between the subject world (where communication takes place) and the object world (the world communicated about). In a transaction, all action takes place in the subject world except for the execution, which takes place in the object world.

Business Action Theory

Business Action Theory is the most recent generic business model that is based on LAP (Goldkuhl, 1996; 1998). The BAT model is intended to be used for business interaction between different organizations, and not for interaction within one organization. This is different from Action Workflow and DEMO, which can both be used for interaction between, as well as within organizations. BAT uses the two roles of customer and supplier and describes the generic interaction that takes place between them. These two roles perform actions directed towards each other that form a pattern. This pattern of interaction is described in six generic business phases (see Fig. 6):

1. Business prerequisites phase
2. Exposure and contact search phase
3. Contact establishment and proposal phase
4. Contractual phase
5. Fulfilment phase
6. Completion phase

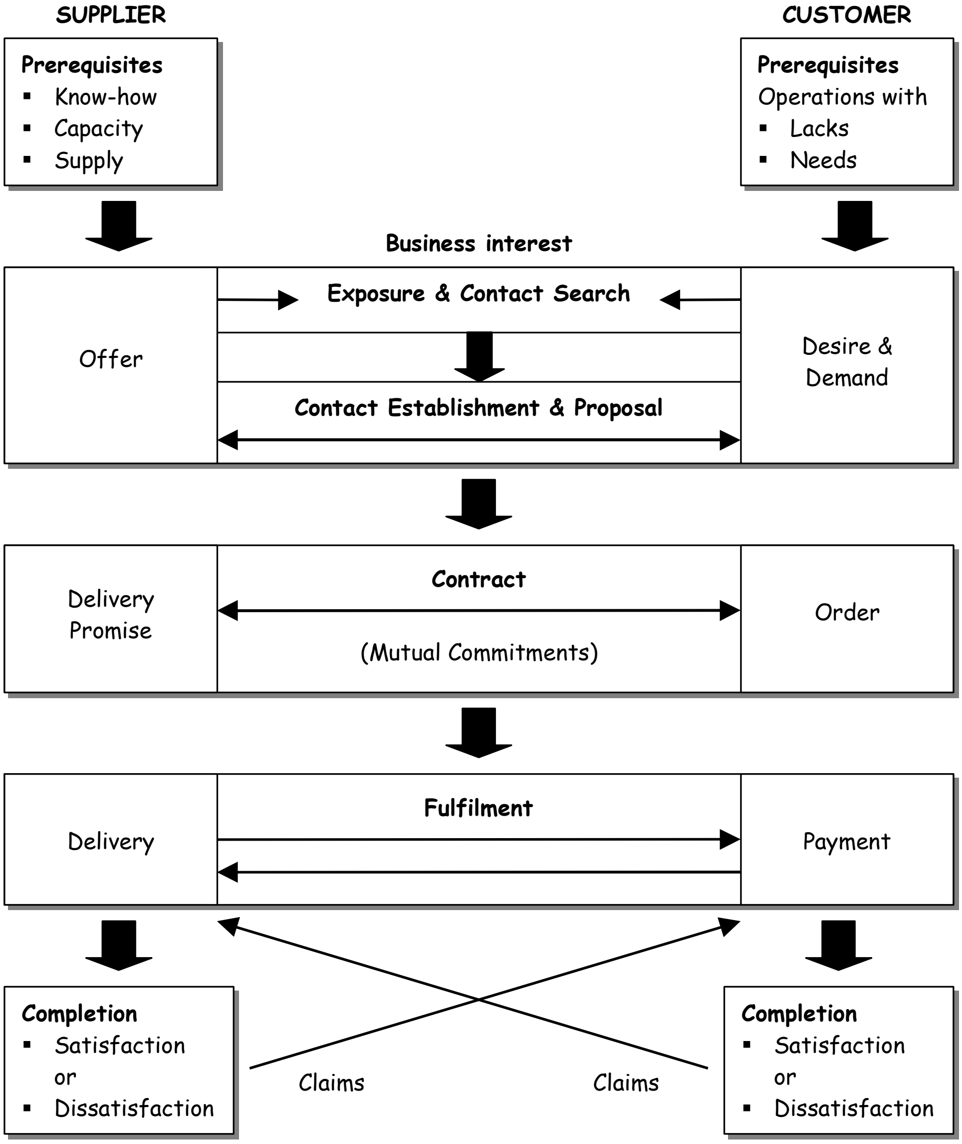


Fig. 6: Business Action Theory: A phase model (Goldkuhl, 1998).

There are similarities in the phases between BAT and Action Workflow. However, BAT is a more exhaustive model that starts earlier and includes material actions such as delivery and payment. The reason for this is that business interaction should be seen as an integrative whole (an 'action game'), consisting of different actions that are both communicative and material.

The first BAT phase is defined as establishing prerequisites (for supplier and customer) for performing business (sales/purchases). On the supplier side, the key word is *ability*. The supplier must have the ability (capacity and 'know-how') to perform the business – to make offers and contracts and to fulfil these contracts. The customer does not have the corresponding ability (or has certain reasons for not utilizing such an ability). In the operations of the customer, there are *lacks and needs* that may be satisfied by potential suppliers and their products (goods/services).

The second and third phases can be viewed together as a *business interest* stage. In the second phase, both parties search for contact. The ability of the supplier is exposed and offered to the market. The lacks and needs of the customer give rise to desire and potential demand, which guide a possible search for products or suppliers. When supplier and customer find each other, they *establish contact* and perhaps start *negotiating* (phase three). The communication here can be described as *proposal stating*. Bids and counter-bids are made. The desires and demands of the customer are expressed and the supplier can make different offers. Of course, in many cases, fixed (and standard) offers must be taken or rejected as made.

The negotiation will transfer into a contractual phase if the proposals are not rejected by either party. Contracting is performed in the fourth phase. The key word here is *agreement*. Customer and supplier come to an agreement concerning the business transaction. The contract is a mutual communicative action expressing the *mutual commitments* made (i.e., commitments for future actions). This involves a delivery promise by the supplier. The customer's order also includes an obligation of future payment. These different commitments must be *fulfilled*. Otherwise, the contract is broken. The supplier must deliver and the customer must pay (phase five).

If not satisfied with the delivery, the customer can make a *claim*. The supplier is requested to make some modification in the delivery. Correspondingly, the supplier can make payment claims towards the customer. This sixth and last phase involves *assessments* of the fulfilment leading to *satisfaction* or *dissatisfaction*.

This generic business interaction model describes the inherent business logic that applies when customers and suppliers perform business with each other. It describes generic business actions of both a communicative and material character. Of necessity, doing business involves communication, otherwise the customer and supplier cannot agree on the business deal. Nevertheless, business interaction cannot be reduced to only incorporate communication. It must include the material acts of delivering goods and/or services and paying.

Essential in a business interaction is the *exchange of value*. Such an exchange of value consists of the *delivery* of a product (goods and/or services) and *payment* being made in return. To be able to exchange value, the supplier and customer must communicate. Such *business communication* includes the exchange of proposals and commitments. All six phases (with the possible exception of phase one) consist of exchanges. Each party directs actions towards the other party. Phase two, exposure and contact search, includes the *exchange of interest*. Seller and buyer signal their possible interest in doing business. In phase three, customer and supplier *exchange proposals*. They communicate bids and counter-bids. They express preferences and try to influence each other in order to arrive at an acceptable deal. In the contractual phase (phase four), the parties *exchange commitments*. They commit themselves to future actions (i.e., the proposed exchange of value). This *exchange of value* takes place in the fifth phase, the fulfilment. The completion phase, the sixth and final phase, can include the *exchange of acceptances or claims*.

Interaction can take place in all of these six phases. The first phase is concerned with the establishment of business prerequisites. This phase does not necessarily involve actions directed towards the other party. In later phases, for example during negotiation (phase three), there might be a need to communicate more thoroughly the prerequisites (the abilities) of either or both parties. This means an *exchange of knowledge concerning business prerequisites* takes place. This is an example of the iterative character of business processes. Later stages can trigger actions belonging to earlier phases. This kind of knowledge exchange concerning preconditions is often important in long-term relationships (i.e., partnerships between supplier and customer).

Interaction means that the two parties act towards each other. Business Action Theory emphasizes that the two parties (business roles) are actively engaged in business interaction. Each party performs actions (decisions, communicating, value exchange) directed towards the other party.

SYSTEMS DEVELOPMENT FROM A LANGUAGE ACTION PERSPECTIVE

In this section, we present and discuss the requirements engineering method VIBA/SIMM. Before doing so, we will introduce another systems development method based on LAP – the Commodious method. As both methods have their foundation in LAP, they share some obvious similarities. However, VIBA/SIMM extends Commodious in several ways, in particular by emphasizing the importance of taking human-computer interaction (HCI) into account early in the development lifecycle. We actually stress that HCI aspects can and should be treated as an important part of business modelling (Ågerfalk *et al.*, 1999).

The Commodious Method

The Commodious method (COMmunication MODeling as an aid to Illustrate the Organizational Use of Software) was developed by Peter Holm and Jan Ljungberg (Holm, 1994; Holm & Ljungberg, 1996; Holm, 1996). The aim of Commodious is to improve the design of software systems supporting organizational communication. The objective of the method is twofold (Holm, 1996, p.11).

1. The use of standard templates and checking mechanisms is assumed to shorten development times and to make it possible to test new ways of working without spending too much time on analysis, requirements specification, implementation, and programming.
2. By introducing a richer and non-technical terminology into the classifications of, for instance, actions, discourses and software modules, the design models are assumed to become easier to understand and control for non-computer experts.

The aim of the first objective is to shorten development schedules, without decreasing the quality of the final product. The second objective reflects an aim towards adopting user-driven participatory design, which is completely in line with the so-called 'Scandinavian approach' (cf. Iivari & Lyytinen, 1998).

Commodious is founded on a combination of the conversation for action schema and a traditional process decomposition mechanism. The purpose of extending the conversation for action schema is to arrive at a generic schema that is editable (situation adaptable) by designers. Another important part of Commodious is its aim to extend the restrictive discourse on organizations implied by speech act theory with the flexibility promoted by conversation analysis (Holm & Ljungberg, 1996). Within speech act theory, communication is supposed to comply with a predefined action structure, which is very much evident in the conversation for action schema. Within conversation analysis, on the other hand, a conversation is made up of pairs of utterances (speech acts) and the parties participating in the conversation are free to change the course of action at any point in time. Regarding high-level business tasks as discourses that are constituted by conversations creates a combination of these two different linguistic schools. Within each conversation, the use of IT artefacts is analysed and classified according to the functional role of IT in relation to social activities. The classification used, and its graphical representation, is shown in Table 2.

The modelling approach proposed by the Commodious method constitutes a two-phase structure. First, the overall business is described in discourse schemas. A discourse schema is a directed graph consisting of the (types of) actions to be performed and the sequence of those actions (i.e., in what order the actions are to be performed). Fig. 7 shows an example of a discourse schema from a paper mill. At the paper mill, a feasibility check is always performed before a customer order is acknowledged. The feasibility check might result in a need to order additional raw material before the availability of material can be declared.

Metaphors for IT-support	Graphical representation
IT as a communication medium	
IT as performance agent	
IT as passive information provider	
IT as active advisor (guidance support)	
IT as tool support	
IT as resource handler	

Table 2: Classification of IT-support according to the Commodious method (Holm & Ljungberg, 1996).

The second step in Commodious is to recognize certain sequences in the discourse to be primitive sequences referred to as conversations. A Commodious conversation corresponds to what we refer to as interactive usage situations. Each identified conversation is analysed in depth and described in conversation schemas. It is within conversation schemas that the classification of IT-support shown in Table 2 is used.

From the same example as in Fig. 7, Fig. 8 shows the conversation needed to determine whether enough raw material is in stock or if new material needs to be ordered. From the conversation of Fig. 8, we see how IT is used as an active advisor, helping to determine the needed amount of raw material, and whether or not that amount is in stock.

Furthermore, we can also see the dual character of the two actions ‘Raw material purchasing’ and ‘Declare availability of material’. Both are performed with the information system as a communication system, and as actions that update the database of the system – the so-called ‘description events’. The box behind the actual action represents this second aspect.

In addition to these models, a ‘system behaviour model’ and an ‘information model’ should be produced. The former specifies ‘software events’ and relates them to the speech acts performed. The latter represents a conceptual model in terms of a system global entity/relationship model that is updated by the specified software events.

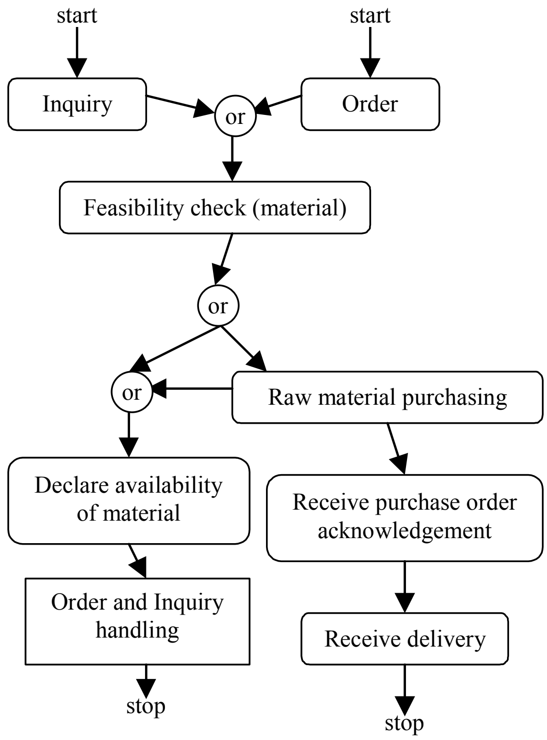


Fig. 7: The 'feasibility check with respect to material' discourse schema at the Paper Mill.

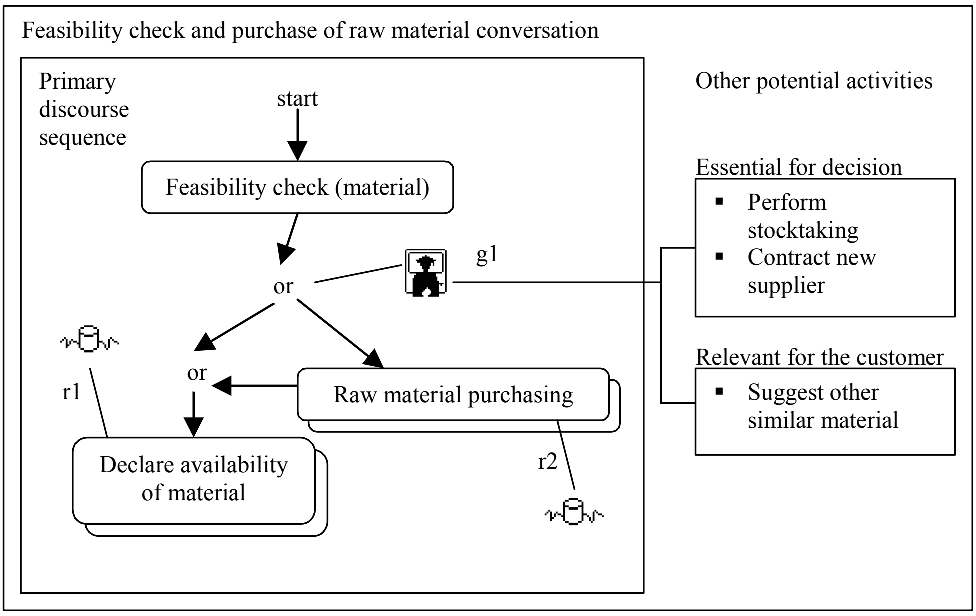


Fig. 8: IT-support in 'feasibility and raw material purchase conversation'.

Versatile Information and Business Requirements Analysis

The Commodious method helps by directing attention to the performance of speech acts within organizations and their relations to information systems. However, its simplified view of organizational discourses as constituted only by speech act sequences, leads to a risk that other important actions are omitted as, for example, material actions that might also be important for the communication (as discussed above). We also argue that the Commodious method lacks a proper understanding of the importance of taking usability into consideration during the early phases of systems development.

Hence, we will now introduce a requirements engineering method that takes the language action perspective even further than Commodious. Essentially, this is achieved in two respects. Firstly, the proposed approach to model business processes covers both language action and material action. It also takes into account the action objects, which are prerequisites for actions as well as results of actions. Secondly, the treatment of interactive usage situations goes into more detail concerning the layout of the interactive screen documents and action sequence restrictions than does the Commodious method.

Requirements engineering according to VIBA/SIMM is carried out within two primary focal areas: Business Process Modelling (BPM) and Information Systems focused Modelling (ISM).

Business Process Modelling

During BPM the business goals, problems, strengths and weaknesses, action structure, and other development constraints are analysed. These activities are often performed as a direct continuation of Change Analysis/SIMM (Goldkuhl & Röstlinger, 1993) or some other feasibility study that is supposed to be conducted before system development starts. The main outcome of BPM is a business action model documented in Action Diagrams. Such diagrams show the action logic of the business, with both the results of, and the prerequisites for, actions.

In Action Diagrams, different actions of a business process and the way these actions are related to each other are explicitly described. Actions performed by human actors, as well as IS actions, are considered. Action Diagrams can be used to describe material flow and information flow within a business process. Material objects and information (messages) are described and related to actions as prerequisites (input) or results (output). One important notational feature is that Action Diagrams describe the performer of each action (i.e., what actor/actor group (role) or which IS is supposed to perform which particular action). Some actions (in Action Diagrams) are delineated to be interactive with several performers (e.g., Customer ↔ Salesman ↔ System).

One important aspect of Action Diagrams is their semantic power to describe action logic. It is possible to describe the sequential order of actions (i.e., the flow aspect), alternative actions (decision points), conjunctive actions, contingent actions (i.e., actions occurring only sometimes), triggering (initiation) of actions (by time or communication), interruption of actions (by time or communication), conditions for actions, and parallel actions.

A contextual descriptive approach is preferable to use when working with Action Diagrams. Each Action Diagram describes a business context within a business process. Different Action Diagrams are related to each other through descriptive connectors (i.e., links to other Action Diagrams). The demarcations of each Action Diagram are arbitrary (i.e., an analyst is free to choose appropriate boundaries for each described context).

Fig. 9 and Fig. 10 show examples of Action Diagrams describing two connected business contexts. See Appendix A for a symbol legend.

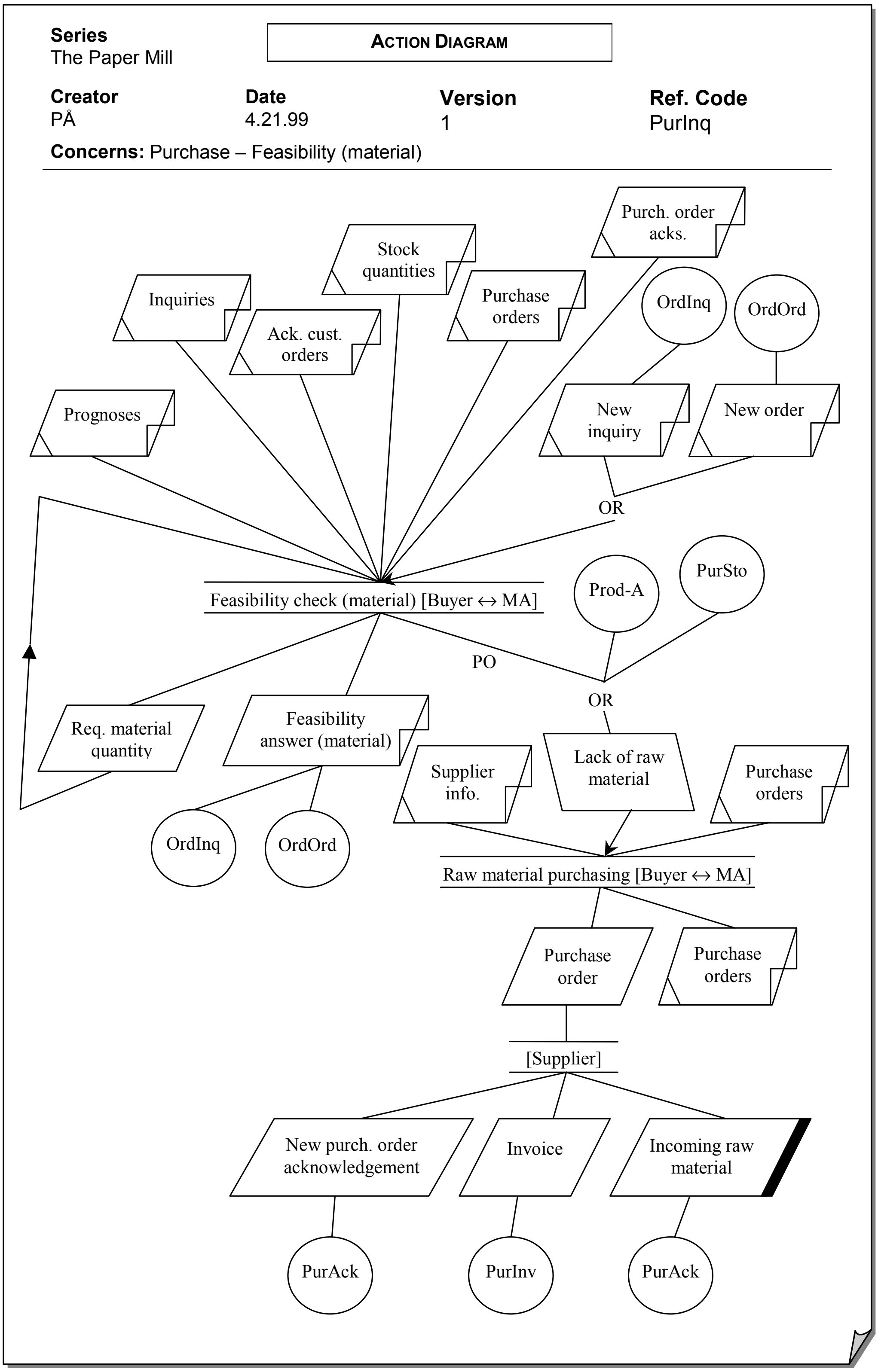


Fig. 9: Example, Action Diagram (I).

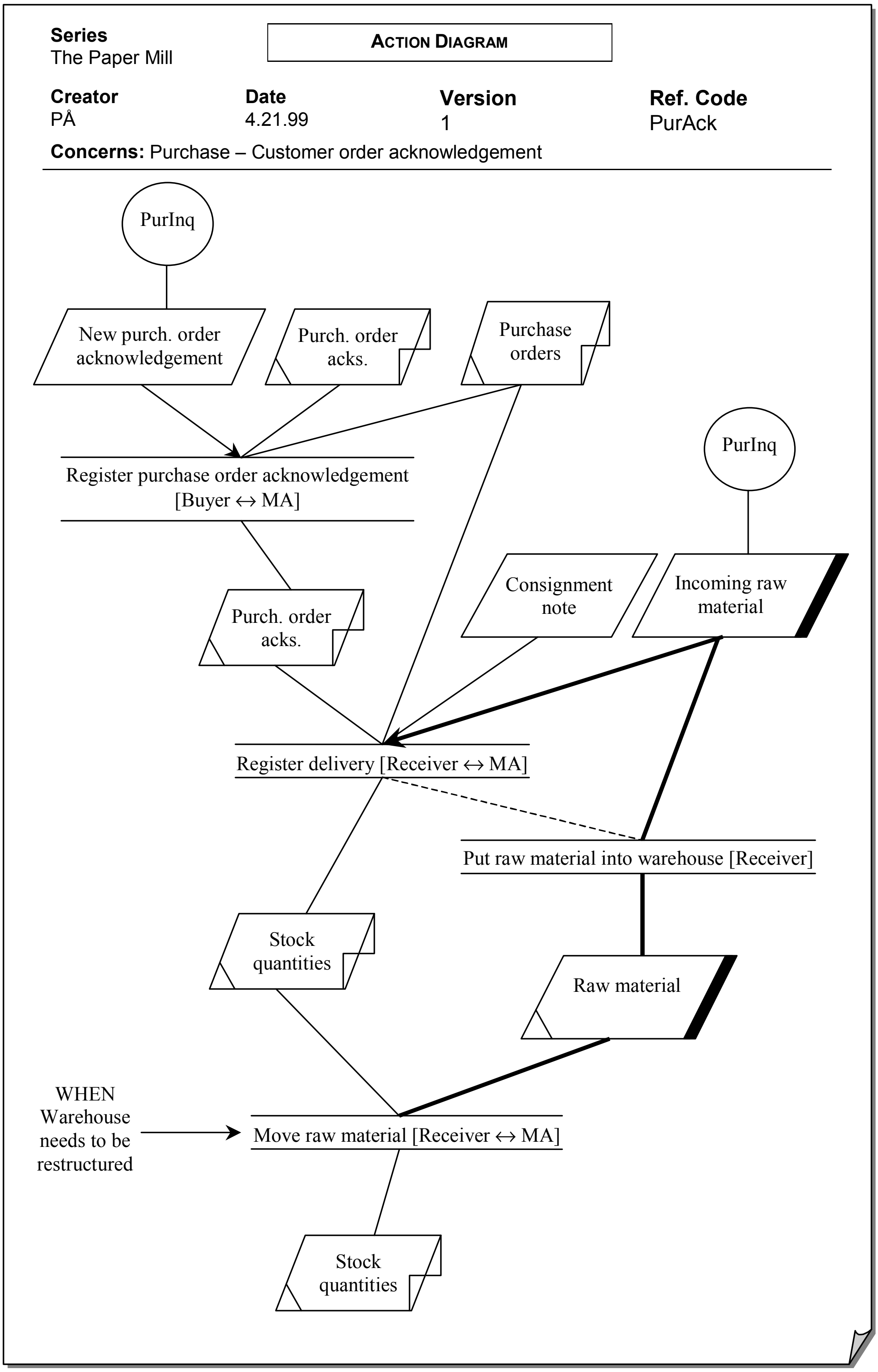


Fig. 10: Example, Action Diagram (II).

Information System Focused Modelling

Information System focused Modelling (ISM) is divided conceptually into three focal areas: Interaction Analysis, Conceptual Analysis, and Document Analysis. However, these three areas are typically treated simultaneously with a continuing shift in focus between them.

Interaction Analysis

To describe the elementary interactions (e-interactions) needed to formulate and send the messages of any given interactive usage situation (ISP), we propose to use Interaction Tables (I-Tables). An I-Table is a table with three rows and three columns. The left-hand column is for the user's actions, the middle column for the state of the current interactive document(s), and the right-hand column is used for the IS actions. One I-Table is used for each e-interaction.

Fig. 11 shows an E-Interaction List, which is the document type used to describe e-interactions. One E-Interaction List is produced for each ISP. An E-Interaction List consists of three parts. Firstly, there is a list of initial IS actions that are to be performed upon entry to the ISP. Secondly, there is a listing of the e-interactions to be performed in the ISP. Finally, there is one I-Table for each e-interaction, which specifies the interaction in detail.

From the I-Table describing e-interaction 3.3 in Fig. 11, we see how s_0, cell (1,2), specifies that the 'Purchase order' document with information about the ID of the current order, the order date and the buyer's reference person need to be visible in order to perform e-interaction 3.3 (i.e., a precondition for the e-interaction). Cell (2,1) specifies what action the I-Table is concerned with, that is, associating a supplier with a purchase order. Cell (2,2) specifies the state of the e-interaction after the user action, and the specific graphical user interface (GUI) components used to perform the e-interaction, in this case, a combo-box labelled 'Sup.ID'. The supplier is now associated with the purchase order (from the user's point of view). Column 3 shows the actions that are to be performed by the IS in response to the user's action. Cell (3,2) shows the response from the IS (i.e., the purchase order document with supplier information shown). Finally, cell (3,1) shows the interpretation act to be performed by the user. In this case, the interpretation should yield either that the correct supplier has been associated with the purchase order or that another supplier must be chosen. The latter case leads to the reperformance of the same (type of) e-interaction.

In most cases, the number of IS actions that corresponds to a user action is one or perhaps two in strict sequence. However, sometimes there might be several IS actions involved and, thus, there is a need to describe how these relate to each other. In such cases, for example, a regular expression specifying the sequence can be added to column 3.

The granularity with which the documents are represented in I-Tables depends on how far the analysis has proceeded. At first, the documents are referred to by textual references (as in Fig. 11). When the analysis proceeds, the layouts of the documents become increasingly explicit. This evolution can (and should) be shown in the I-Tables by inserting thumbnails of documents or document parts in column 3.

It is sometimes unnecessary to show the initial state s_0 within an I-Table and, hence, the first row can be omitted, as shown in I-Tables 3.1 and 3.2 of Fig. 11.

Series
The Paper Mill

E-INTERACTION LIST

Creator	**Date**	**Version**	**Ref. Code**
PÅ	4.26.99	4	EI3

Concerns: ISP 3 – Purchasing

Initial system actions

S3.1 Retrieve purchase order with highest Order ID.
S3.2 Show info from order in all applicable fields.
S3.3 Fill cbPurOrderId with all available purchase order IDs in descending order.
S3.4 Fill cbSupId with all supplier IDs in ascending order.

E-interactions

3.1 Create new purchase order.
3.2 Retrieve earlier registered purchase orders.
3.3 Associate supplier with purchase order.
3.4 Print purchase order.
3.5 Register purchase order.
3.6 Change {Date, Our reference, PurOrderLine (dbgOrderLines)}.
3.7 Change {Your reference, Delivery details, Payment details}.
3.8 Cancel new purchase order.

3.1	User action	Document	IS action
S_1	**Create new purchase order**	**Purchase order** "New order" (btnNewOrder)	1. Clear all fields. 2. Generate new purchase order ID (next consecutive) and show in cbPurOrderID. 3. Show today's date in txtDate.
S_2	**State fact:** New purchase order created.	**Purchase order** Order ID (cbPurOrderID) Date (txtDate)	

3.2	User action	Document	IS action
S_1	**Retrieve earlier registered purchase orders**	**Purchase order** "Order ID" (cbPurOrderID)	1. Retrieve order with selected ID. 2. S3.2.
S_2	**State fact:** Selected purchase order retrieved and showed.	**Purchase order** All order information visible	

3.3	User action	Document	IS action
S_0		**Purchase order** Order ID, Date, Our reference	
S_1	**Associate supplier with purchase order**	**Purchase order** "Sup. ID" (cbSupID)	Retrieve information about selected supplier and show in all supplier-related fields.
S_2	**Interpretation:** 1. Correct supplier associated. 2. Wrong supplier associated. (→ 3.3)	**Purchase order** All supplier information visible.	

Fig. 11: Example E-Interaction List (page 1 of 3 in a set of documentation).

The language used within I-Tables is a natural language such as English or Swedish. We do, however, use some conventions that semiformalize the language use, as described in Table 3.

Convention	Example
Constraints regarding the performance of user actions are written surrounded by curly brackets in cell (2,1).	{By drag 'n drop 2→1} means that the user action is performed by use of drag 'n drop from document 2 to document 1 as numbered in cell (1,2).
Constraints regarding the performance of IS actions are written surrounded by curly brackets in column 3.	{RT ≤ 3s} means a response time equal to or less than three seconds.
Clickable items are GUI components used to execute the user action. These are surrounded by quotation marks in cell (2,2).	'New order' is a clickable item labelled 'New order' used to create a new order.
References to names given to GUI components in a prototyping tool might be shown after their 'visible' caption as shown in the GUI.	'Order ID' (cbPurOrderID) means that the clickable item 'Order ID' is named cbPurOrderID in the prototyping tool. This also indicates that it is a combo-box due to the 'cb'-prefix (using Hungarian notation).
To reference a sequence of IS actions, the function Sys can be used.	Sys(3.1) means that the IS actions stipulated in I-Table 3.1 shall also be performed here.

Table 3: Notational conventions used in I-Tables.

Each I-Table describes one e-interaction, so there is a need to describe how these are related to each other. The action potential of any given interactive screen document might vary over time. Since action sequence restrictions constitute the state of the document, and thereby restrict the action potential, we use the formalism of Statecharts (Harel, 1984) to model them. A related issue is the modelling of navigation paths between documents, which could be achieved with I-Tables. However, such an attempt would not be tractable due to the many possible navigation paths. Instead, this is also achieved with Statecharts, where each high-level state represents a document, and each navigational action is represented by a transition. The resulting state model can be viewed as consisting of documents as high-level states where the action sequence restrictions of each document (Fig. 12) correspond to sub-states of those.

This approach to model action sequence restrictions is an extended version of Horrocks' (1999) Statechart approach to user interface construction. With Horrocks' approach, there is a clear path from interface design to implementation in, preferably, event-driven rapid development tools (e.g., MS Visual Basic and Borland Delphi). However, the approach lacks a clear connection to business actions, which is provided by the use of e-interactions as a source for transitions, rather than Horrocks' more vaguely defined notion of event.

Although our approach is an extension of Horrocks' (*ibid.*), the techniques and help-documentation suggested and thoroughly discussed by Horrocks (*ibid.*) might assist when performing Interaction analysis.

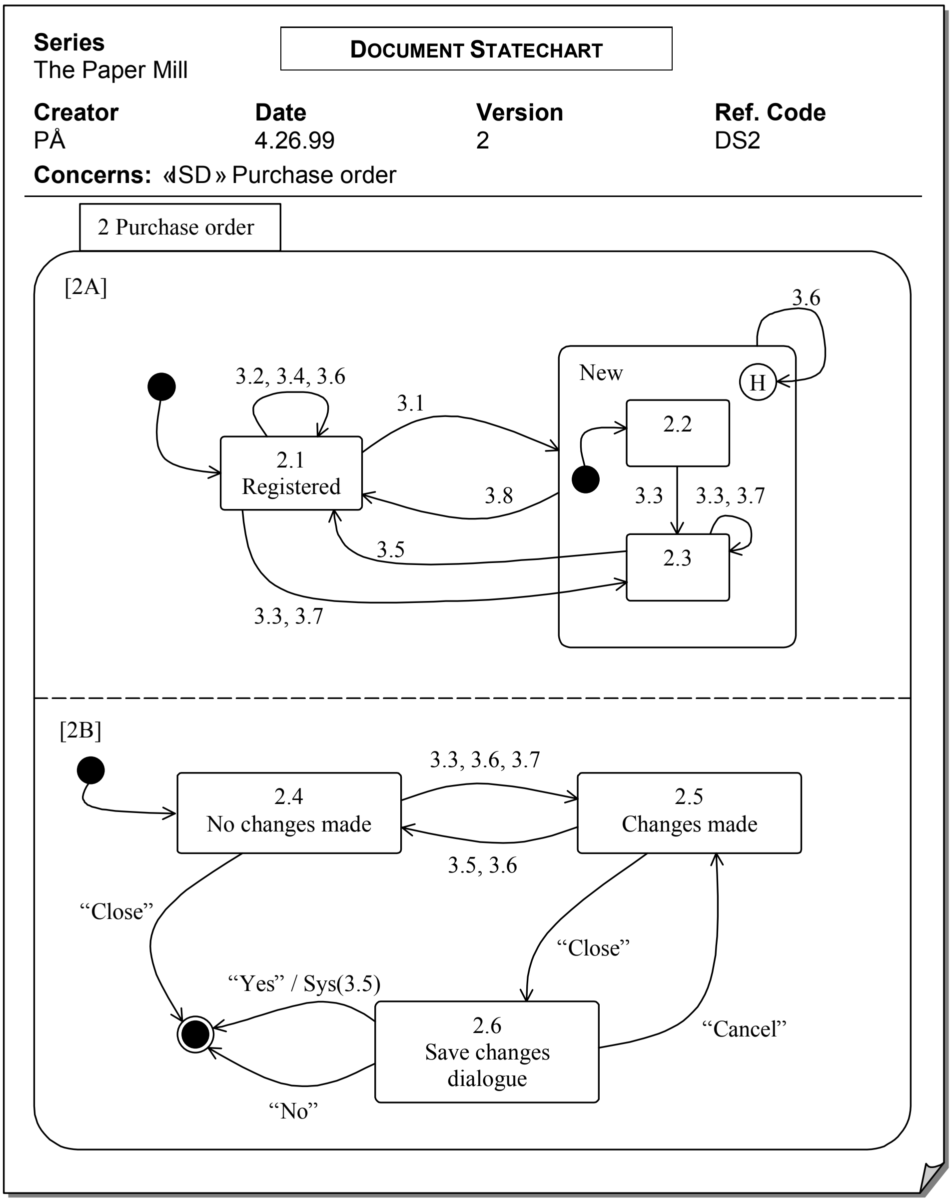


Fig. 12: Example Document, Statechart (II).

Conceptual Analysis

In order to design interactions and interactive documents as well as automatic actions, there is a need to consider the propositional content of messages. This is conducted preferably through a conceptual analysis based on identified messages. Conceptual analysis and Interaction analysis (and Document analysis, see below) are thus performed as an integrated process. During Interaction analysis, the professional language used by actors is captured and used in interactive documents. This language is then analysed by 'traditional' conceptual modelling in terms of classes and properties, (cf. object-oriented analysis, e.g., Graham, 1998; Booch, 1994; Blaha & Premerlani, 1998). To document conceptual relationships we propose the use of the Unified Modelling Language (UML) class diagrams (e.g., Booch *et al.*, 1999), where one class diagram is produced for each (type of) message.

Each class discovered is described in a class definition, which, following UML consists of a brief description, the cardinality of the class (i.e., how many instances are allowed and/or supposed), the class' position in a possible inheritance hierarchy by stating superclasses; attributes, and associations.

Note that the same class might be used in different ae-messages. The Class definition, however, is a 'central' description of the properties of a class that holds for all ae-messages. Hence, not all attributes and associations defined in the Class definition are shown in all Class diagrams.

Following the UML, it is possible to attach an additional state machine to a class in order to describe its dynamic behaviour. This is not always necessary but, if so, a reference to the Class Statechart showing the state machine should be included in the Class definition.

Document Analysis

When designing documents – interactive as well as non-interactive – during Interaction analysis, the focus is on the particular ISP under investigation. Some documents, however, are used in many ISPs and so there is a need to verify that no inconsistencies arise. One way to ensure this is to create a Document definition for each document, as shown Fig. 13.

The heading of a Document definition contains the 'usual' meta-information, including the name of the defined document. The name is prefixed by a stereotype stating what kind of document we are dealing with, for example, a paper document, or an interactive screen document. The tentative classification we have used so far is shown in Table 4. This naming convention is used throughout the documentation whenever a document is referenced (for example, in Fig. 12).

A Document definition consists of six sections. First, there is a verbal description of the document briefly describing its intended use. Subsequently, there are references to message(s) carried by the document, message(s) created by the document, document(s) created by the document, which are usually paper documents, the ISPs in which the document is used, and, finally, a prototype of the document.

VIBA/SIMM encourages the use of prototypes in conjunction with analytical modelling (Goldkuhl & Ågerfalk, 1998). A prototype is a visualization of a document, usually created in parallel with I-Tables and Statecharts.

Series
The Paper Mill

DOCUMENT DEFINITION

Creator	**Date**	**Version**	**Ref. Code**
PÅ	4.29.99	4	DD2

Concerns: «SD» Purchase order

Description: A Purchase order interactive screen document is used to formulate and send purchase orders (MD2). Physically, purchase order paper documents (DD2P) are printed by use of this document and subsequently sent to suppliers.

Message(s) carried: MD2 – Purchase order

Message(s) created: MD2 – Purchase order

Document(s) created: DD2P – «PD» Purchase order

Used in ISP(s): ISP 3 – Raw material purchasing

Prototype: DP2

Fig. 13: Example, Document Definition for an interactive screen document.

Stereotype	Meaning
«PD»	Paper Document
«ISD»	Interactive Screen Document
«SSD»	Static Screen Document (i.e., non-interactive)

Table 4: Stereotypes used for different types of documents.

As we can see, most of the activities performed during Document analysis are actually performed intertwined with Interaction analysis. The reason for treating it as a separate focal area is that it promotes a holistic view of the documents used, whereas Interaction analysis is focused on one ISP at a time.

CONCLUSIONS

In this paper, we have highlighted a need to consider the action character of information when modelling information in business processes. We have presented the Language Action Perspective (LAP) as the perspective of choice for information modelling – now and for the future – if actable information systems are to be constructed. As examples of generic business frameworks founded in LAP, we have presented Action Workflow, DEMO, and Business Action Theory. The latter we propose as the framework to be used when applying the requirements engineering method VIBA/SIMM, which we have outlined and positioned towards the (somewhat similar) Commodious method. See Ågerfalk (1999) for a more thorough description of VIBA/SIMM and its underlying theory of information systems.

Although LAP has become a mature theoretical orientation with obvious practical importance and applications, it is important to realize that much still remains to be achieved. Commodious and VIBA/SIMM, for example, are important steps towards the operationalization of LAP into method support for information modelling and systems development. However, more work is needed – both theoretical and empirical. One important future research topic is, for example, further elaboration on the concept of actability; a concept that we believe will eventually play an even larger role than the restricted (and perhaps outdated) concept of usability when it comes to usage of information technology in the future.

REFERENCES

Action Technologies (1993). *Action Workflow Analysis Users Guide*. Action Technologies.

Ågerfalk P J (1999). *Pragmitazation of Information Systems – A Theoretical and Methodological Outline*. Licentiate Thesis, Linköping University, Sweden.

Ågerfalk P J, Goldkuhl G, Cronholm S (1999). Information Systems Actability Engineering: Integrating Analysis of Business Processes and Usability Requirements. In Goldkuhl, Lind, Seigerroth and Ågerfalk (eds.), *Proceedings of 4th International Workshop on the Language Action Perspective on Communication Modelling (LAP'99)*. Copenhagen, September 12-13, 1999, pp. 73-86.

Ågerfalk P J, Goldkuhl G. (1998). Elicitation and Analysis of Actability Requirements. In Fowler & Dawson (eds.), *Proceedings of 3rd Australian Conference on Requirements Engineering*. Geelong, Victoria, Australia, Oct 26-27, 1998.

Auramäki E, Lehtinen E, Lyytinen K (1988). A speech-act-based office modeling approach. *ACM Trans of OIS*, vol. 6, no 2, pp. 126-152.

Austin J L (1962). *How to do things with words*. Cambridge, MA: Harvard University Press.

Blaha M, Premerlani W (1998). *Object-oriented modeling and design for database applications*. Englewood Cliffs, NJ: Prentice Hall, Inc.

Booch G (1994). *Object-oriented analysis and design with applications.* 2nd ed. Redwood City, CA: Benjamin/Cummings Publishing Company, Inc..

Booch G, Rumbaugh J, Jacobson I (1999). *The Unified Modeling Language User Guide*. Addison Wesley Longman, Inc.

Chen P (1976). The entity-relationship model – toward a unified view of data. *ACM TODS* 1, No. 1, March 1976.

Cronholm S, Ågerfalk P J, Goldkuhl G (1999). From Usability to Actability. *Proceedings of 8th Intl. Conference on Human-Computer Interaction (HCI International'99)*, vol. 1, pp. 1073-1077. Munich, Germany, Aug 22-27, 1999.

Denning P J, Medina-Mora R (1995). Completing the loops. *Interfaces*, vol. 25, no. 3, pp. 42-57.

Dietz J (1994). Business modelling for business redesign. In *Proceedings of the 27th Annual Hawaii International Conference on Systems Science*, IEEE.

Flores F, Ludlow J (1980). Doing and speaking in the office. In Fick & Sprague (eds.), *Decision support systems: Issues and challenges*, Pergamon Press.

Goldkuhl G (1995). *Information as Action and Communication*. In Dahlbom (ed.), *The Infological Equation: Essays in Honor of Börje Langefors*, Gothenburg: Gothenburg Studies in Information Systems.

Goldkuhl G (1996). Generic business frameworks and action modelling. *Proceedings of Workshop on Communication modelling – the Language/Action Perspective* (LAP'96), 1996, Springer Verlag.

Goldkuhl G (1998). *The Six Phases of Business Processes – Business Communication and the Exchange of Value*. JIBS Working Paper Series, Jönköping International Business School. Presented at Beyond Convergence: The 12th Biennial ITS Conference (ITS'98) in Stockholm, Sweden.

Goldkuhl G, Ågerfalk P J (1998). Action Within Information Systems: Outline of a Requirements Engineering Method. *Proceedings of 4th Intl. Workshop on Requirements Engineering: Foundation for Software Quality*. Pisa, Italy, June 8-9, 1998, pp. 133-154.

Goldkuhl G, Lyytinen K (1982). A language action view of information systems. In Ginzberg & Ross (eds.). *Proceedings of 3rd International Conference on informations systems*, Ann Arbor, December 13-15, 1982.

Goldkuhl G, Röstlinger A (1993). Joint elicitation of problems: An important aspect of change analysis. In Avison D *et al*. (eds.) *Human, organizational and social dimensions of Information systems development*, North-Holland/IFIP w.g. 8.2.

Graham I (1998). *Requirements Engineering and Rapid Development: An Object-oriented Approach*. ACM Press, Harlow: Addison Wesley Longman Limited.

Habermas J (1984). *The theory of communicative action 1. Reason and the rationalization of society*. Beacon Press.

Holm P (1994). The Commodious Method – COMmunication MODeling as an aid to Illustrate the Organizational Use of Software. *Proceedings of 6th Intl. Conference on Software Engineering and Knowledge Engineering*. Jurmala, Latvia, 1994.

Holm P (1996). *On the Design and Usage of Information Technology and the Structuring of Communication and Work*. Doctoral dissertation. Stockholm University, Sweden.

Holm P, Ljungberg J (1996). Multi-Discourse Conversations. In *Proceedings of 4th European Conference on Information Systems (ECIS'96)*, Lisbon.

Horrocks I (1999). *Constructing the User Interface with Statecharts*. Addison-Wesley.

Iivari J, Lyytinen K (1998). Research on Information Systems Development in Scandinavia – Unity in Plurality. *Scandinavian Journal of Information Systems*, vol. 10, no. 1&2, pp. 135-186.

Jacobson I, Ericsson M, Jacobson A (1995). *The object advantage: business process reengineering with object technology*. ACM-press.

Langefors B (1966). *Theoretical Analysis of Information Systems*. 4th ed., Studentlitteratur, Lund, Sweden, 1974.

Lundeberg M, Goldkuhl G, Nilsson A (1981). *Information systems development – A systematic approach*. Englewood Cliffs, NJ: Prentice-Hall.

Searle J R (1969). *Speech Acts. An Essay in the Philosophy of Language*. Cambridge University Press, Cambridge.

Searle J R (1979). *Expression and Meaning. Studies in the Theory of Speech Acts*. Cambridge University Press, Cambridge.

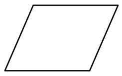
van Reijswoud V E (1996) *The structure of business communication. Theory, model and application*. Doctoral dissertation, Technical university, Delft, The Netherlands.

Whitaker (1992). *Venues for Contexture*. Doctoral dissertation. Umeå University, Sweden.

Winograd T, Flores F (1986). *Understanding Computers and Cognition: A new Foundation for Design*. Ablex, Norwood.

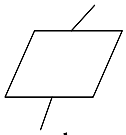
Yourdon E (1989). *Modern structured analysis*. Prentice Hall Inc, Englewood Cliffs, New Jersey.

APPENDIX A – SYMBOLS USED IN ACTION DIAGRAMS

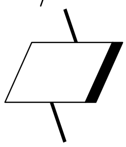


Information (oral or written) action object.

Material action object.



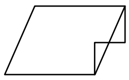
Information flow (communication).



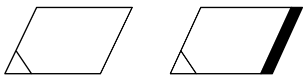
Material flow.

Activity [Performer]

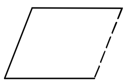
Activity with named performer.



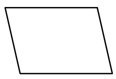
Encapsulated information.



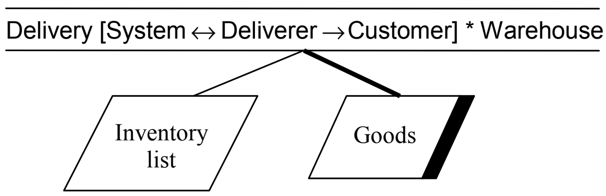
Store of action objects. Applicable to both information and material.



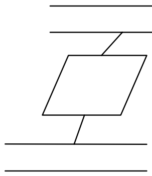
Knowledge, which has not been externalised.



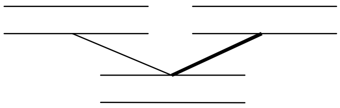
Non-action. An action object that is the result of an omitted action (i.e. an omission).



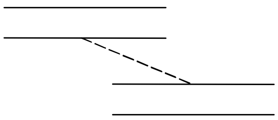
Activity with several performers, named location and resulting action objects.



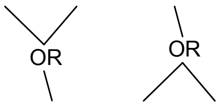
Ordered sequence of activities with action object as result and prerequisite respectively.



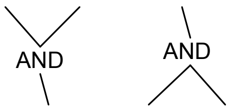
Suppressed action object. To be used only if the meaning is clear from the context, to increase readability.



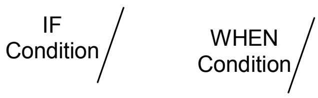
Ordered sequence of activities with no intermediate action objects.



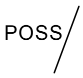
Alternative.



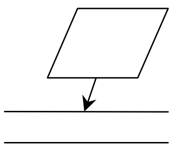
Conjunction.



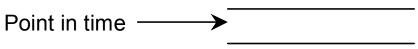
Condition for action or action object. Often used in combination with alternatives and combinations.



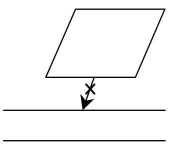
Occasional action or action object ('POSS' is an abbreviation of 'possibly')



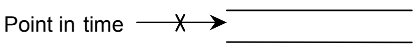
Activity triggered by communication.



Activity triggered by time.

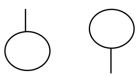


Activity interrupted by communication.

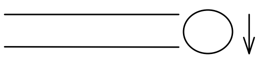


Activity interrupted by time.

Parallel activities.



Sequential connector to/from another Action diagram. The Ref. Code of the connected diagram is shown in the circle.



Hierarchical connector to another Action diagram that shows a decomposition of the connected activity.