WORKING PAPER I-2008-04

WP

Lars Bækgaard & Peter Bøgh Andersen

Using Interaction Scenarios to Model Information Systems

Informatics
Research Group

USING INTERACTION SCENARIOS TO MODEL INFORMATION SYSTEMS

Lars Bækgaard[1]

Peter Bøgh Andersen[2]

ABSTRACT

The purpose of this paper is to define and discuss a set of interaction primitives that can be used to model the dynamics of socio-technical activity systems, including information systems, in a way that emphasizes structural aspects of the interaction that occurs in such systems. The primitives are based on a unifying, conceptual definition of the disparate interaction types – a robust model of the types. The primitives can be combined and may thus represent mediated interaction. We present a set of visualizations that can be used to define multiple related interactions and we present and discuss a number of case studies that indicate that interaction primitives can be useful modeling tools for supplementing conventional flow-oriented modeling of business processes.

KEYWORDS

Information systems. Interaction. Scenarios. Modeling. Socio-technical systems.

1 Introduction

Interaction is a widely used concept and occurs in many different areas of information systems and information systems development. Information systems can be viewed as activity systems (Checkland and Holwell 1998) or work systems (Alter 2006) in which human beings interact with other human beings and with technology and objects. DEMO (Dietz 2006) and BAT (Goldkuhl 1996; Goldkuhl and Lind 2004) are business models that view business activity in terms of interaction between business parties.

We present and discuss an approach to interaction modeling that is based on a set of interaction primitives and a corresponding visualization technique that supports modeling of complex networks of interactions. We are not aware of any existing modeling approach that supports such a variety of interactions as is covered by our primitives.

The underlying rationale is that interaction is a fundamental characteristic of information systems and that interaction modeling should play a substantial role in information systems analysis and design. Each individual interaction in an information system can be viewed as a dynamic relation between an actor and one or more elements in the system. Interaction is a source of internal and external change. Exchange of representations between two elements in an information system is a source of internal change. Exchange of representations between an information system and its environment is a source of external change in the relations between these two entities.

---

[1] Aarhus School of Business, University of Aarhus, Fuglesangs Alle 4, DK-8210 Aarhus V, Denmark, e-mail: lab@asb.dk.

[2] Department of Information and Media Studies, University of Aarhus. Helsingforsgade 14, DK-8200 Aarhus N, Denmark, e-mail: pba@imv.au.dk.

Within the area of human-computer interaction, the concept is understood as interaction between human beings and computers (Rogers, Sharp et al. 2002). Use cases represent systems that offer services to actors (Cockburn 2001). A use case specification defines the interaction between a system and one or more actors; it does not define the interaction between the actors unless their interaction is mediated by the system. Neither human-computer interaction nor use cases can capture general interaction between two human actors.

Many modeling languages support activity modeling. However, each of them supports a limited form of interaction modeling. UML interaction diagrams represent interaction by messages through which objects control objects (Rumbaugh, Jacobson et al. 1999). Data flow diagrams represent interaction by data flows that enable two activities to interchange representations (De Marco 1978). Activity diagrams (Rumbaugh, Jacobson et al. 1999), EPC diagrams (Dehnert 2002; Lübke, Lüecke et al. 2006), and BPMN diagrams (White 2004) can be used to represent two different types of interactions: exchange of representations between activities and transfer of control between activities.

Clearly, interaction is an important concept that is analyzed in restricted and somewhat ad hoc ways. In order to resolve this problematic situation, we propose a set of interaction primitives that cover all the types of interactions that are inherent in the above-mentioned approaches. We supplement the primitives with a visualization technique that makes it possible to model situations which are characterized by multiple related interactions.

Our research method can be characterized as design science (March and Smith 1995; Hevner, March et al. 2004). The design aspect is represented by our interaction primitives and the corresponding visualization technique. The science aspect is represented by two case studies that we use as a basis for an evaluation of our modeling approach.

The paper is structured as follows. In Section 2 we analyze a number of modeling languages in order to identify the types of interaction modeling that they support. In Section 3 we define an ontology that serves as a conceptual foundation for our interaction primitives. In Section 4 we present the interaction primitives and show how they can be combined to model mediation. In Section 5 we present a visualization technique that can be used to model activity systems in terms of multiple related interactions. In Section 6 we present two cases that we use to evaluate our modeling technique. In Section 7 we present the conclusions of the paper and suggest directions for future work.

2 Activity and action

The purpose of this section is to identify a basic set of action types, which we call interaction primitives. Briefly, an interaction primitive can be viewed as a pattern that defines a dynamic relation between two elements. One of the elements performs an action. We use activity theory to qualify actions as occurrences within activities. And we use the notion of activity systems to characterize some of the structures that occur when activities are executed. We analyze a set of modeling languages in order to identify the types of interaction primitives that they support. The interaction primitives can be combined into interaction scenarios that represent a static view on the dynamics of activity systems.

2.1 Activity systems

An activity system can be viewed as a system where actors transform something in order to create value for customers within a specific environment (Checkland 1981; Alter 1999). Business activities deal with movement, manipulation, and consumption of material objects and information objects and they deal with coordination in terms of requests for, agreements about, control of, and evaluation of work activities (Denning and Medina-Mora 1995).

Activity theory can be used to qualify the activities that are performed by activity systems (Leontiev 1978). An activity is carried out by a set of actions performed by actors that have a shared purpose with the activity. An action is a conceptualization of something that an actor does. For example, "a customer sends an order" can be perceived as an action. Each action is executed in terms of a set of operations. Hence, "a customer sends an order" may be carried out in terms of operations like "a customer opens an order application" and "a customer presses the submit button". Actors that modify objects with or without mediating tools are a central theme in activity theory.

An action may be characterized by a combination of material and communicative aspects (Winograd and Flores 1986; Goldkuhl 2001). For example, a document-based customer order has material aspects in terms of the physical movement of the document and it has communicative aspects in terms of the intended meaning of the symbols on the document.

Information systems can be viewed as activity systems that are responsible for information activities. They register, store, manipulate, and present information about a domain of interest to actors in support of their activities (Checkland and Holwell 1998). Information systems play important roles in material activities and coordination activities. Material activities may be mediated by digitally controlled machinery, and many coordination activities are communication activities in which actors express requests, requirements, contracts, and evaluations.

Information systems exist within larger activity systems that involve manipulation of both things and information. Increasingly, information systems become tied to things and actors by means of chips that are attached to these things and actors. These chips may contain information about the corresponding things and their history: a bottle of milk may contain information about the origin and production of the milk, and a parcel may contain information about its destination and purpose. This implies that things play important roles for and in information systems even though information systems do not contain things. Information systems process information about activity systems (and other relevant phenomena). This may include information about actors, things, logical relations, and interaction within and outside the activity system of which the information system is a part.

We define the term interaction as a set of actions that occur as two or more elements affect each other. The idea of mutual effect is essential, as opposed to a one-way causal effect. Thus, we define the term interaction primitive as a type of action whose instances can be combined with other instances (of an interaction primitive) in order to create interaction.

2.2 Botanizing modeling languages

In this section we discuss three modeling languages with respect to their use of interaction primitives. We describe the basic idea of each language and the types of interaction primitives that it supports. This botanizing gives us four basic kinds of interaction primitives that are useful for designing information systems.

A data flow diagram can represent interaction in terms of flows of representations between participants (De Marco 1978). A participant can be an external source/consumer of representations, a representation store, or an activity that manipulates representations. The partial data flow diagram in Figure 1 represents a situation where a representation r flows from activity a1 to activity a2.

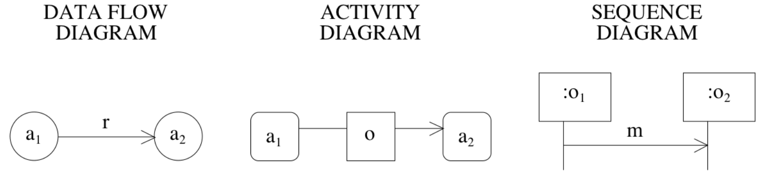

Figure 1 Interaction in modeling languages

A UML activity diagram can represent interaction in relation to flows of objects between activities and one activity controlling another activity (Rumbaugh, Jacobson et al. 1999). The partial activity diagram in Figure 1 represents a situation where an object o flows from activity a1 to activity a2.

A UML sequence diagram can represent interaction in terms of messages that are passed among objects (Rumbaugh, Jacobson et al. 1999). The partial sequence diagram in Figure 1 represents a situation where one object o1 sends a message m to another object o2. A message is a request that activates an action in the receiving object. Messages can contain parameters.

Sensing is a type of interaction where a participant senses aspects of someone or something. For example, a customer may listen to a radio in an electronics store. The data flow diagram in Figure 1 can be interpreted as a sensing where the activity a2 senses the representation r if r is loosely coupled to its medium. This situation may for instance occur when a copy of a digital file is transferred via a network. After the transfer, both a1 and a2 have access to r. The activity diagram in Figure 1 can be interpreted as a sensing where the activity a2 senses the object o. The sequence diagram in Figure 1 can be interpreted as a sensing where the object o2 senses parameters that are passed via the message m.

Moving is a type of interaction where something or someone is moved from a source to a destination. For example, an employee may move items from a storage room to a shelf in a store. The data flow diagram in Figure 1 can be interpreted as a moving where the representation r is moved from activity a1 to activity a2 if r is physically bound to its medium. This situation may for instance occur when r is physically bound to a piece of paper or a digital medium. The activity diagram in Figure 1 can be interpreted as a moving where the object o is moved from activity a1 to activity a2. The activities in an activity diagram may be explicitly located in terms of named swim lanes. The sequence diagram in Figure 1 should not be interpreted as a moving. The reason is that the parameters that are passed via the message m are per definition not coupled to a medium.

Controlling is a type of interaction where one participant controls the behavior of another participant. For example, an accountant may control a piece of accounting software in order to get certain computations done. The activity diagram in Figure 1 can be interpreted as an instance of controlling where activity a1 terminates itself and initiates activity a2 when the object o flows from a1 to a2. The sequence diagram in Figure 1 can be interpreted as controlling where the object o1 controls the actions of the object o2 by sending a message that initiates a certain set of actions in o2.

Modifying is a type of interaction where one participant modifies something. For example, a programmer may modify a piece of source code in order to add new functionality. The sequence diagram in Figure 1 can be interpreted as an instance of controlling where the object o1 controls the actions of the object o2 by sending a message m that initiates a certain action in o2. Modifying can also be indicated in activity diagrams since the exchanged objects may have visible states that change as they are passed from activity to activity.

This survey has provided us with four types of interactions that are scattered in a number of modeling methods: sensing, moving, modifying, and controlling. In the following we present of model of these interactions and ways of visualizing them. The model is given in the form of a class hierarchy.

2.3 Discussion

The actions that underlie our four interaction primitives have both material and communicative aspects in so far as they involve signs. The material aspect of a sign is its representation whereas the communicative aspect is captured by the notion of reaction and information. In the formal notations that follow below, we mostly describe the material side, i.e. the representation. The reason is that both human actors and IT systems can handle this aspect. When it comes to reaction and information, the two actors differ. The main difference is that the rules that govern the reaction of IT systems are fixed, whereas the habits governing the human reaction are negotiable and mutable. This implies that the material actions performed by IT systems cannot be guided by unforeseen interpretations of the situation at hand unless one or more human actors participate and use their interpretations and judgments to guide the IT systems.

3 The model

In this section we build a conceptual ontology for the analysis. Wherever possible, we have looked for theoretical or empirical underpinnings of the ontology in order not to bother the reader with yet another homespun ontology.

An entity is a chunk of space-time and an event is an entity with defined temporal boundaries (Russell and Norvig 2003). Events may involve other entities such as participants with certain roles (Tesnière 1959; Fillmore 1968; Fillmore 1977; Halliday 1994). Using this terminology, “the Second World War” and “my birthday” are examples of events.

A role is a standardized, linguistically codified function which participants can choose in specific events (Fillmore 1968; Fillmore 1977). Table 1 shows the roles which we will be using in the following. The third column indicates the grammatical construction normally used to signal the role in sentences and the type of participant that can fill it.

Role	Definition	Realization
Agent	The active participant that initiates and controls the event	Subject: entity
Object	The passive participant that is most affected by the event	Object: thing
Content	The description of the contents of the communication	Object: event
Experiencer	The participant affected by information about a phenomenon	Subject: thing
Phenomenon	That which is thought, felt, or sensed by the experiencer	Object: entity
Beneficiary	The participant to whom something is given or for whom something is done	Indirect object, adverbial: thing
Addressee	The intended experiencer of a message	Indirect object, adverbial: thing
Instrument	The passive participant that enables the event	Adverbial: thing
Location	The spatial boundary of the event	Adverbial: place
Time	The temporal boundary of the event	Adverbial: interval
Source	The location from which an object is transported	Adverbial: place
Destination	The location to which an object is transported	Adverbial: place

Table 1. A definition of roles and their realization. The roles listed above the bold line are actants, the roles below it are circumstances

Roles can be divided into actants and circumstances. Actants are obligatory roles (Tesnière 1959; Halliday 1994): Agent, Object, Content, Experiencer, Phenomenon, Beneficiary, Addressee. Actants are used to characterize the interaction primitives in Section 4. Circumstances are facultative roles (Tesnière 1959; Halliday 1994): Instrument, Location, Time, Source, Destination.

An interval is an event that is only defined by fixed temporal boundaries: today, in the twenty-first century, my birthplace is an entity only defined by fixed spatial boundaries: Aarhus, Denmark (Russell and Norvig 2003).

A thing is an entity with defined but possibly changing spatial boundaries and defined temporal boundaries. An entity like a product has a spatial boundary against its surroundings and it begins and ceases to exist at a certain time. The distinction between events and things is a matter of perspective since language can treat events as things. If, for instance, a race is viewed at close quarters, we would say, "The horses raced for five minutes". Seen from afar, we would turn the verb into a noun and say, "The race was moved from Aarhus to Copenhagen". In the latter case, the event is given a spatial boundary and therefore classified as a thing that can be moved.

An activity is an event consisting of a set of actions united under a common long-term motive activity: maintaining customer satisfaction, ensuring quality control. An action is an event with an effect on other actions and consisting of operations: reading a file, driving a car, flying to Copenhagen, pealing potatoes (Leontiev 1978). A process is an action composed of identifiable sub-events without a termination criterion (Vendler 1967). The sub-events belong to the same category as the process of which they are part (liquid events: taking a stroll, eating apples, collecting data, visiting customers (Russell and Norvig 2003)). An accomplishment is an action composed of sub-events with a termination criterion (Vendler 1967). A sub-event does not belong to the same category as the accomplishment of which it is part (non-liquid events: flying to Copenhagen, making mashed potatoes (Russell and Norvig 2003)).

A state is an action in which something remains constant (Vendler 1967; Russell and Norvig 2003): sleeping, standing at the sea, lying in the grass. A state change is a momentary action composed of two consecutive, different states (Vendler 1967): waking up, winning the lottery. Some OOAD methodologies treat all actions as state changes, i.e. momentary actions with no duration (Mathiassen, Munk-Madsen et al. 2000). This is probably sufficient for administrative systems where you are only interested in recording the fact that an action has been performed. However, in process control, the system must record the temporal evolution of the process; so state changes will not suffice here.

Bindings: a description of the propensity of the participant to fill a certain role (cf. Valences in (Tesnière 1959)). We will use four numerical variables: ability, desire, obligation, and right (Andersen 2006). Bindings are used to deal with the fact that actions may be suboptimal or conflict ridden. For example, a pump may be worn down and only partially able to participate in the action of circulating the cooling water or the operator may be obliged but not particular willing to supervise a plant.

Effect: participating in actions changes the participants' relation to other actions by decreasing or increasing the bindings that connect them to other actions. We thus describe the effect of an action through the new action possibilities that it opens or closes. For example, by using, a wrench the garden owner is enabled to replace the sparking plug of the mower. By reading the time and place of a meeting in the calendar, an employee is enabled and obliged to participate in the meeting. Lind identifies four types of effects: an action can produce, maintain, destroy, or suppress another action (Lind 1994).

An actor is an entity that is able to, desires, is obligated, or has a right to participate in actions as Agent or Experiencer. Since this is a purely functional definition, humans as well as technical systems may be actors in relation to a particular action. However, they may not be equally suited for all actions. For instance, a clock is a good Agent in the action The clock ticks whereas it would be a very bad Agent in the action The clock sent an order confirmation. It therefore makes sense to talk about IT systems and software components as playing the roles of Agent or Experiencer in a specific action. Both things and events can be actors for the reasons given above: the distinction seems to be a matter of the observer's perspective.

A sign is a triadic relation between a representation, the object represented, and the reaction produced by the observer or producer of the sign: a document, a screen image, a warning signal, an utterance, a database, a computational object.

Information is the reaction selected by the observer or producer of a sign as compared to the range of possible reactions. Luhmann describes communication as consisting of a number of selections made by the speaker and the listener (Luhmann 1984): the listener in particular selects an understanding and an action. Examples: answering a doorbell as opposed to ignoring it; confirming an order as opposed to canceling it. The concept of information used here is different from that of information theory, but it retains the idea of selecting among a set of possibilities, and the idea of reducing uncertainty. Luhmann would say that communication reduces complexity (Luhmann 1984). An information system is thus a system that processes representations with the purpose of changing the repertoire of actions of its users (cf. above).

An interaction primitive is an action involving at least two actants. One of them must be an Agent or an Experiencer. For instance, the system printed a document for the user; the system fetched data from the database; the employee read the sales statistics.

Mediation is two interaction primitives A and B sharing at least one participant, called the mediator. A is subordinate to B. Participating in A enables or obligates the mediator to participate in B. For example, the radio receiver on my roof picks up signals from the sender at the local school (A) and this enables it to be an instrument of my Internet communication (B).

An activity system is a standardized set-up of interaction primitives that realize a recurrent activity. Examples: a library, including the cataloging and circulation control system, or a power plant, including the control room.

4 A model of interaction primitives

In Section 2 we found four interaction primitives by botanizing in various modeling languages. In this section we define an ontology-based model of the primitives. The model is not a modeling language, but rather a source of various visualizations and notational forms suited for a variety of purposes.

Our notion of interaction is based on a basic set of unidirectional interaction primitives. As argued in the previous section, we view interaction as a dynamic relation between two elements in an activity system. This implies, for example, that we view the (modifying) actions of an actor that modifies an object as interactive actions. Similarly, we view the (observing) actions of an actor that observes an object as interactive actions. The primitives can be combined to represent bidirectional interactions.

We do not claim that our four interaction primitives constitute a complete set of primitives that cover all imaginable forms of interaction. They were collected from existing methodologies and new technologies may appear that necessitate new primitives. For example, pervasive computing emphasizes space and movements in space in a way that we have not seen in existing methods (Bardram and Bossen 2005).

4.1 Interaction primitives

SENSE is a primitive that represents a situation where an Experiencer senses aspects of a Source. Its actants are Experiencer, Phenomenon, and Source. It can be a process or an accomplishment. Its effect depends upon the Phenomenon sensed: reading a book increases the Experiencer's ability to talk about it, whereas reading a warning sign decreases his desire to progress further.

SENSE has two variants. The general variant of SENSE represents a situation where the Experiencer is different from the Source. Example: a person may listen to music from a radio. FEEL is a variant that represents a situation where the Experiencer is identical to the Source. Example: a person may sense aspects of his own emotional state.

Experiencer	Senses	Phenomenon	Source
He	hears	music	from the radio

He	sees	her coming	
He	smells	the odor	of the fish
He	reads	a copy	of the file
The sensor	reads	the rfid tag	of the book

Table 2. The SENSE primitive

MOVE is a primitive that represents a situation where an Agent moves an Object from a Source to a Destination. The actants are Agent, Object, Source, and Destination. MOVE is a process or an accomplishment. The effect is to increase the Object's ability to participate in actions whose Source equals the Destination of the move action. Example: flying from Copenhagen to Stockholm enables a person to fly from Stockholm to Madrid.

MOVE has four variants. The general variant (MOVE) represents a situation where the Agent is different from both the Source, the Object, and the Destination. Example: a customer can transport products from a store to his home. GIVE represents a situation where the Agent is identical to the Source. The Agent/Source gives an object to a Destination. Example: a customer can give an order to an employee. TAKE represents a situation where the Agent is identical to the Destination. The Agent/Destination takes an Object from the Source. Example: a customer can take a product from a shelf. WALK represents situations where the Agent is identical to the Object. The Agent/Object moves itself from a Source to a Destination. Example: an employee can walk from an office to a department store.

Agent	Moves	Object	Source	Destination
The customer	throws	the product		into the basket
The pump	circulates	the water	from the heater	to the cooler
The train	carries	the passenger	from Aarhus	to Copenhagen
The function	reads	data	from the file	into the buffer
The function	sends	Information	from the server	to the client

Table 3. The MOVE primitive

MODIFY is a primitive that represents a situation where an Agent modifies an Object. The actants are Agent and Object. Examples: persons can change the properties of things; persons can combine things into new things; persons can divide things into groups of things; employees can modify raw materials into products; programmers can modify software; journalists can write articles; IT systems can use their actuators to modify objects. MODIFY is a process or an accomplishment. The effect solely depends upon the kind of modification. For instance, assembling a chair enables the assembled chair to participate as Destination in the action of sitting down, whereas cooking potatoes enables them to participate as the Object of eating.

Agent	Event	Object
The cook	makes	a pizza
The software agent	changes	the user profile
The database module	deletes	a row in the database

Table 4. The MODIFY primitive

CONTROL is a primitive that represents a situation where an Agent uses requests to control an Experiencer or where an Agent physically controls an Object. The actants are Agent and Experiencer. Examples: a department manager asks an employee to undertake a certain task; a business intelligence system asks for specific representations in a database; actors initiate activities; actors redirect a flow of activities; actors suspend and terminate activities.

Requests can be communicative or materiel depending on the qualities of the Agent and the Experiencer. Both the Agent and the Experiencer must be human beings in order for a request to be communicative. When one person requests something from another person, the request may take the form of a linguistic expression such as, "Please, give me the butter". A person that controls a user interface may express a request in linguistic terms such as, "Select all employees from New York". However, the request is material because the user interface is bound to react to the request in a material manner.

Agent	Event	Experiencer
The system	selects rows from	a database
The boss	requests a report from	an employee
The user	turns on	the system
The reading method	is called	the constructor method
The user interface	terminates	the simulation

Table 5. The CONTROL primitive

An IT system can be controlled by a set of requests that the IT system can respond to. Such a set may include requests like trigger, pause, resume, terminate, etc. The requests of a controlled IT system define an action space for the controlling actor. The action space depends on the request set and the IT system's responses to each request. A more comprehensive account of human flexibility is outside the scope of the present paper and belongs to the study of text linguistics and conversation analysis.

The CONTROL primitive is a state change. The effect is to increase the Experiencer's obligation to act as the Agent of the Content slot.

4.2 Mediation

Mediation consists in two interaction primitives A and B sharing one or more participants, called mediators M. When M participates in A, it is enabled or obligated to participate in B. Mediated actions can be verbalized by leaving out the mediator and letting the main actor be the Agent of the action. For example, CONTROL can be combined with MOVE as follows: the customer requests the website to send an order to the company; the captain requests the first officer to order the helmsman on deck.

Experiencer	Senses	Phenomenon	(Source)
The user	observes	customer behavior	from the **system**
The **system**	records	customer behavior	from the customer

Table 6. Mediated SENSE

Interaction can be direct or it can be mediated by persons, IT systems, tools, machines, etc. Direct interaction includes contact between the involved elements (actors, things, representations). For instance, when an employee moves a product by hand, the interaction between the employee and the product is direct. The interaction between an employee and a customer is direct when they interact face to face.

Mediation involves a mediator (actor or object) between the interacting parties. For example, when an employee uses a truck to move a product, the interaction between the employee and the product is mediated by the truck. When an employee exchanges representations with a customer in a chat room, the interaction between the employee and the customer is mediated by the chat room. Mediation introduces new interactions between actors and mediators. When two actors communicate in a chat room, they must interact with the chat system in order to interact with each other. The original interactions between the actors are the primary interactions. The interactions between actors and mediators are secondary interactions that are introduced as a consequence of the mediation of the original interactions.

IT systems can be used to mediate interaction. For example, word processing software mediates interaction between a writer and a text. Chat software mediates interaction between the communicating actors. E-commerce software mediates shared events in terms of business transactions that involve businesses, products, and customers.

Mediation plays an important role in material and coordination activities. Material activities may be mediated by digitally controlled machinery and many coordination activities are communicative actions in which actors express requests, requirements, contracts, and evaluations.

5 Views

In the following we introduce a set of graphical visualizations for our four interaction primitives. The primitives constitute a model that can be visualized in several ways, depending upon the purpose. The golden rule is that the difficult and problematic parts of the representations are represented by pictorial means (arrows etc.) whereas the unproblematic parts are represented by text or left out.

An important choice is the level of detail. Components of word meanings can be atomic building blocks, as in Schank's conceptual dependency diagrams (Dunlop 1990). Whole word meanings can be building blocks, as in Sowa's conceptual graphs (Sowa 2000). Sentences can be building blocks when we want to represent chains of events and illustrate how events influence one another (Fillmore 1968; Fillmore 1977; Dik 1989; Nurcan, Etien et al. 2005; Andersen 2006). We have created our notation to support modeling of complex networks of interactions. Diagrams are only useful as abstractions when things get complicated. Simple situations need no diagrams.

5.1 Interaction primitives

The present proposal distinguishes between participant representations and role representations. Each type of interaction is represented by a different type of arrow that implicitly defines the roles associated with it.

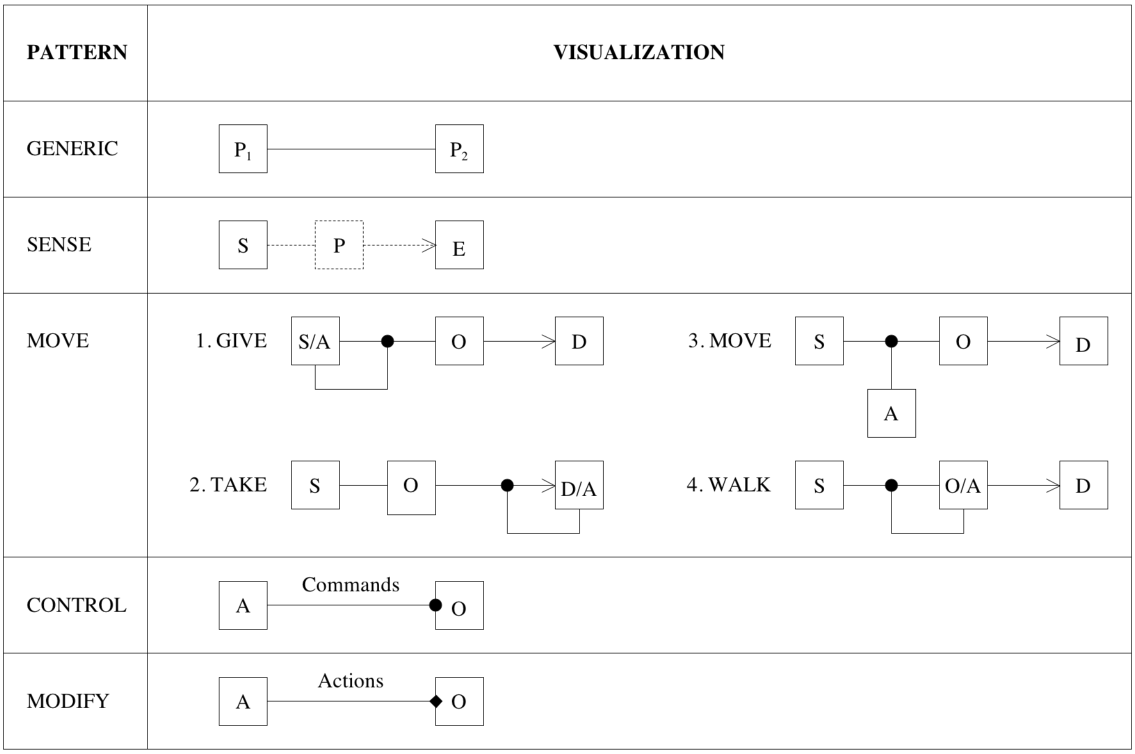

Figure 2 Visualized interaction primitives

In Figure 2 we have shown our notation. Boxes symbolize participants and arrows symbolize interactions and the roles played in these. Participants can be things or events, for the reasons explained in Section 2, and things include representations.

GENERIC is an unspecified interaction primitive that shows two interacting participants (P1, P2) but contains no assumptions about the type of interaction and the associated roles. GENERIC can be used in situations where the specific characteristics of an interaction are not yet clear.

SENSE is a pattern where an Experiencer (E) senses a Phenomenon (P) that is assumed to be a characteristic of a Source (S). The convention is that the Phenomenon is described as it is or should be experienced by the Experiencer, not by the Source. It is thus a receiver-oriented conception of communication. This is indicated by the fact that the arrow runs from the Phenomenon to the Experiencer.

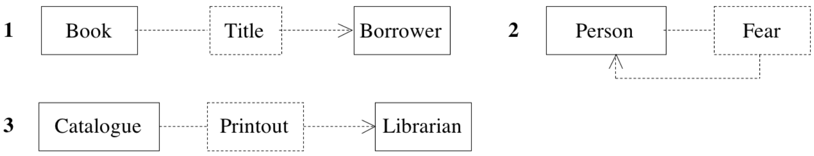

Figure 3 The SENSE primitive – Examples

Figure 3 contains two examples of SENSE primitive. In example 1 a borrower senses (reads) the title of a book. The title is the Phenomenon and the book is the Source. In example 2 a person senses his own fear. This example is a special form of SENSE where the experiencer is identical to the source.

We call this SENSE variant FEEL. In example 3 a librarian senses (reads) a printed copy of a computerized catalogue.

MOVE is a primitive where an Agent (A) moves an Object (O) from a Source (S) to a Destination (D). The arrow runs from the Source to the Destination. The box on the arrow represents the moved Object, for example a thing or a representation.

As illustrated in Figure 2, MOVE has four sub-primitives each of which has a distinct Actor that performs the move action. In each sub-primitive this Actor is related to the move by an arrow. 1. The Source performs the move action. 2. The Destination performs the move action. 3. An external Agent performs the move action. 4. The moved Object performs the move action.

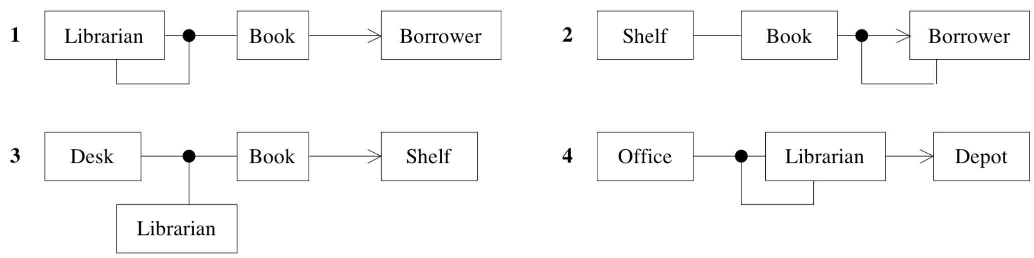

Figure 4 The MOVE primitive – Examples

Figure 4 contains four examples of MOVE primitive. In example 1 a librarian (Source) gives a book to a borrower. In example 2 a borrower (Destination) takes a book from a shelf. In example 3 a librarian (external Agent) carries a book from a desk to a shelf. In example 4 a librarian (Object) goes down into a depot.

CONTROL is a primitive where an Agent (A) influences the actions of an Object (O) by giving requests to the Object. The arrow runs from the Agent to the Object.

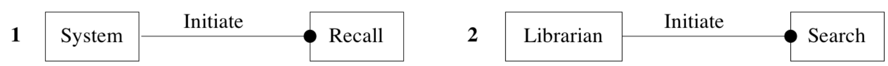

Figure 5 The CONTROL primitive – Examples

Figure 5 contains two examples of CONTROL primitive. In example 1 a system initiates a recall process. In example 2 a librarian initiates a search process.

MODIFY is a primitive where one Agent (A) performs actions with the result that an Object (O) is changed. The Agent acts in a way that changes the Object. The arrow runs from the Agent to the Object.

Figure 6 The MODIFY primitive – Examples

Figure 6 contains two examples of MODIFY primitive. In example 1 a librarian catalogues a book. In example 2 a librarian destroys a book.

5.2 Pre- and postconditions

Each interaction primitive has a general precondition that must be satisfied before an instance of the primitive can occur. Also, each primitive has a general postcondition that must be satisfied after an instance of the primitive has occurred.

PATTERN	INTERACTION	PRECONDITION	POSTCONDITION
SENSE	<Experiencer> senses <Phenomenon> on <Source>	The <Source> can produce the <Phenomenon> The <Experiencer> can sense the <Phenomenon>	The state of the <Experiencer> has changed in response to the <Phenomenon>
MOVE	<Agent> moves <Object> from <Source> to <Destination>	The <Agent> has access to the <Object> that is present at the <Source> The <Agent> has access to a location at the <Destination> that can hold the <Object>	The <Object> is present at the <Destination>
CONTROL	<Agent> imposes <Command> on <Object>	The <Agent> can execute the <Command> The <Object> can respond to the <Command>	The <Object> has responded to the <Command>
MODIFY	<Agent> imposes <Action> on <Object>	The <Agent> can perform the <Action> The <Object> can respond to the <Action>	The state of the <Object> has changed as a consequence of the <Action>

Figure 7 Preconditions and postconditions

The complete set of preconditions and postconditions are defined in Figure 7. These conditions are general because they apply to all instances of the primitives. Specific pre- and postconditions can be defined for specific specializations of the primitives.

SENSE: The precondition states that the Source must be able to produce the sensed Phenomenon and that the Experiencer must be able to sense it. The postcondition states that the state of the Experiencer must be changed in response to the sensing.

MOVE: The precondition states that the Agent must have access to the Object that is to be moved and that the Agent must have access to a location at the Destination to which the Object is to be moved. The postcondition states that the Object must be present at the Destination after it has been moved.

CONTROL: The precondition states that the Agent must be able to execute the Command and that the Object must be able to respond to the Command. The postcondition states that the Object must be able to respond to the Command after it has been performed.

MODIFY: The precondition states that the Agent must be able to perform the Action and that the Object must be able to respond to the Action. The postcondition states that the Object must be changed as a consequence of the Action after it has been performed.

These preconditions and postconditions ensure that the roles Source and Experiencer are played by entities that have certain capabilities. An Experiencer that senses a visual Phenomenon must have visual sensing capabilities. An Agent that moves something must be able to access the moved Object and actually move it. An Agent that imposes a Command must be able to do this and the Object must be able to respond.

5.3 Composition and mediation

Interaction primitives can be combined to form complex interaction scenarios.

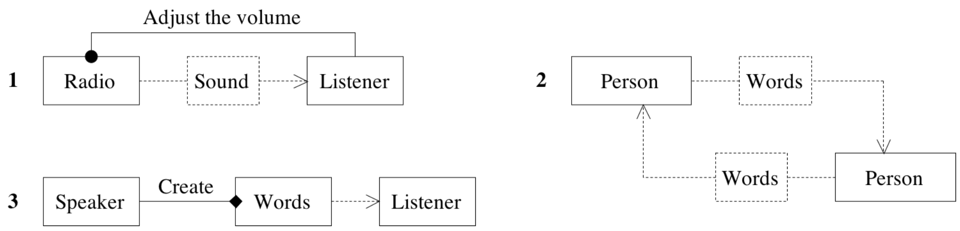

Figure 8 Combined interaction primitives

Figure 8 contains three examples of combined interaction primitives. In example 1 a listener senses sound from a radio. In example 2 two persons sense each other's words. This example shows how interaction primitives can be combined into a model interaction where two or more elements affect each other. In example 3 a listener senses words that are created by a speaker.

Combinations of interaction primitives can be used to create mediation patterns, as illustrated in Figure 9.

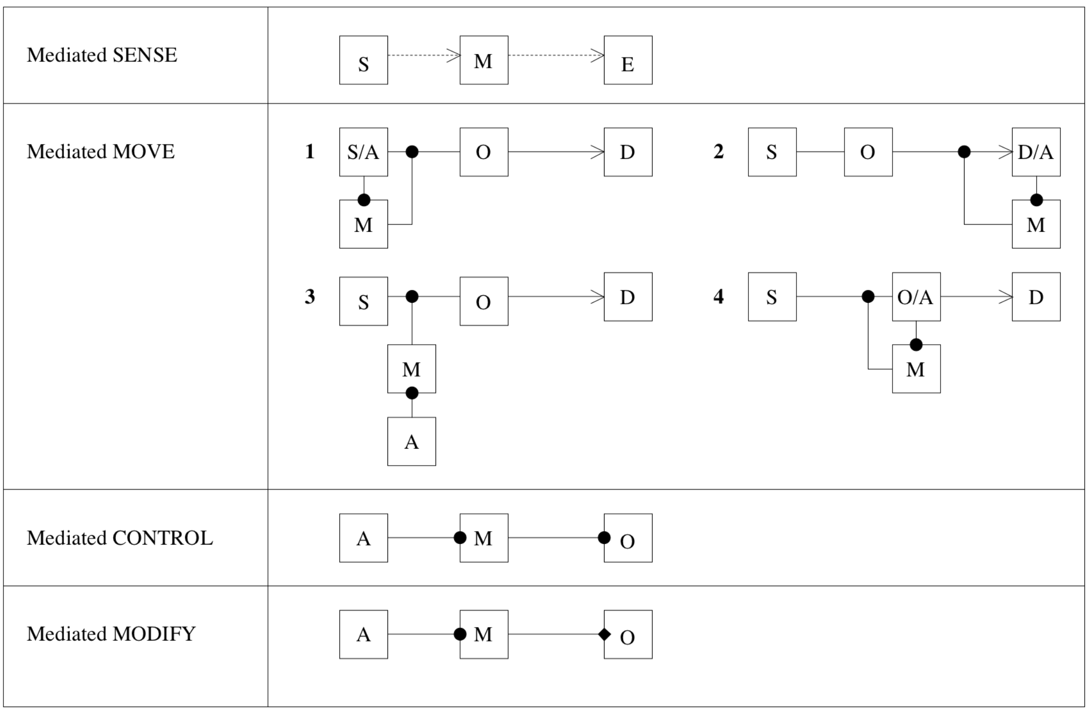

Figure 9 Mediation patterns

Mediated SENSE is a pattern where a Mediator (M) is placed between the Source (S) and the Experiencer. The pattern can be viewed as two connected SENSE primitives.

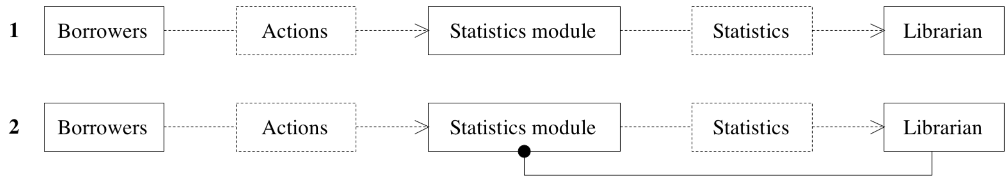

Figure 10 Mediated SENSE – Examples

Figure 10 contains two examples of mediated SENSE. In example 1 a librarian reads statistics that are generated by a statistics module that records (senses) information about library borrowers' actions. The librarian senses aspects of the borrowers indirectly via the statistics module. The statistics module senses aspects of the borrowers directly in terms of their actions. In example 2 the librarian can control the statistics module by means of requests that influence the generated information about the borrowers' actions.

Mediated MOVE is a pattern where a Mediator (M) is placed between the Agent (A) and the move action. This enables the Agent to perform the move action via the Mediator. Like its unmediated counterpart, mediated MOVE has four variants.

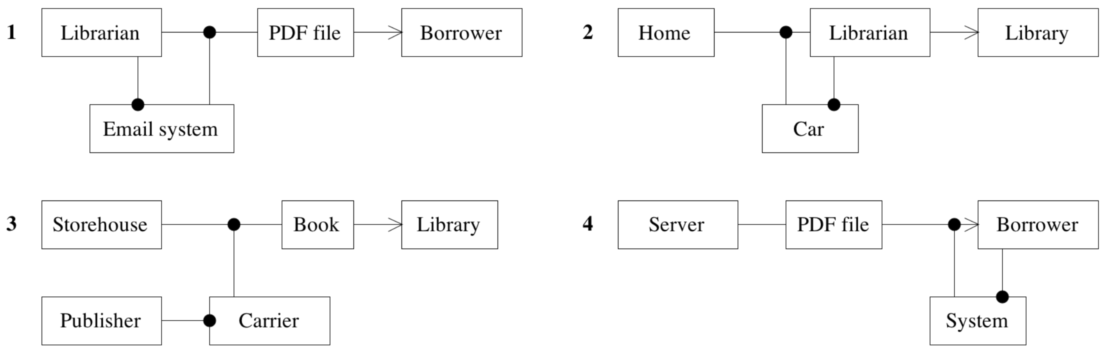

Figure 11 Mediated MOVE – Examples

Figure 11 contains four examples of mediated MOVE. In example 1 a librarian gives a PDF file to a borrower indirectly via an email system. The librarian uses requests to control the Mediator that gives the borrower access to the PDF file. In example 2 a librarian uses a car to move himself from a home to a library. The librarian uses requests like wheel turning to control the car. In example 3 a publisher uses a carrier to move a book from a storehouse to a library. The publisher uses requests to control the carrier. In example 4 a borrower uses a system to get a PDF file from a server. The borrower uses requests to control the system.

Mediated MODIFY is a pattern where a Mediator (M) is placed between the Agent (A) and the Object (O).

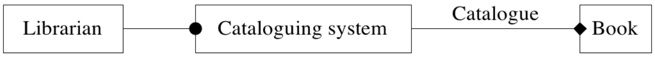

Figure 12 Mediated MODIFY – Example

Figure 12 contains an example of mediated MODIFY. A librarian modifies the status of a book indirectly via a cataloguing system (Mediator). The librarian controls the cataloguing system that modifies the book status.

Mediated CONTROL is a pattern where a Mediator (M) is placed between the Agent (A) and the Object (O).

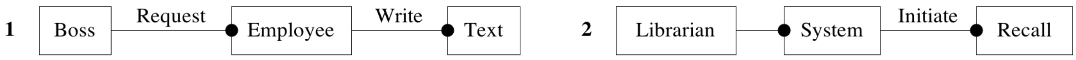

Figure 13 Mediated CONTROL – Examples

Figure 13 contains two examples of mediated CONTROL. In example 1 a boss gets a text written via an employee. In example 2 a librarian initiates a recall process via a mediating system.

It should be clear from our examples that CONTROL is a very important primitive if we want to understand mediation. The notion of a controllable participant that responds to selected requests is inevitable if we want to model mediated interaction.

6 Cases

This section describes two activity systems using the visualizations of interaction primitives from the previous section.

6.1 Beggars and philanthropists

Beggars and philanthropists is a simulator that represents a theory of budgeting. The simulator can be used by researchers to study the behavior of two actor roles: the beggar and the philanthropist. The simulator object, which controls the roles beggars and philanthropists, also controls the transport of statistical data to diagrams. The diagrams are read by political science researchers as trend curves and histograms. The simulator object also reads parameter values from parameter objects that can be modified by the researchers.

The beggars observe their chances of receiving funding from the nearby philanthropists, focus on the most promising ones, and send an application. The philanthropists assess their familiarity with the individual beggar and, depending upon the outcome of the application, issue funding. Beggars and philanthropists are observed by statistics objects that record aggregated counts of their properties and actions. The data is recorded after each round of the simulation.

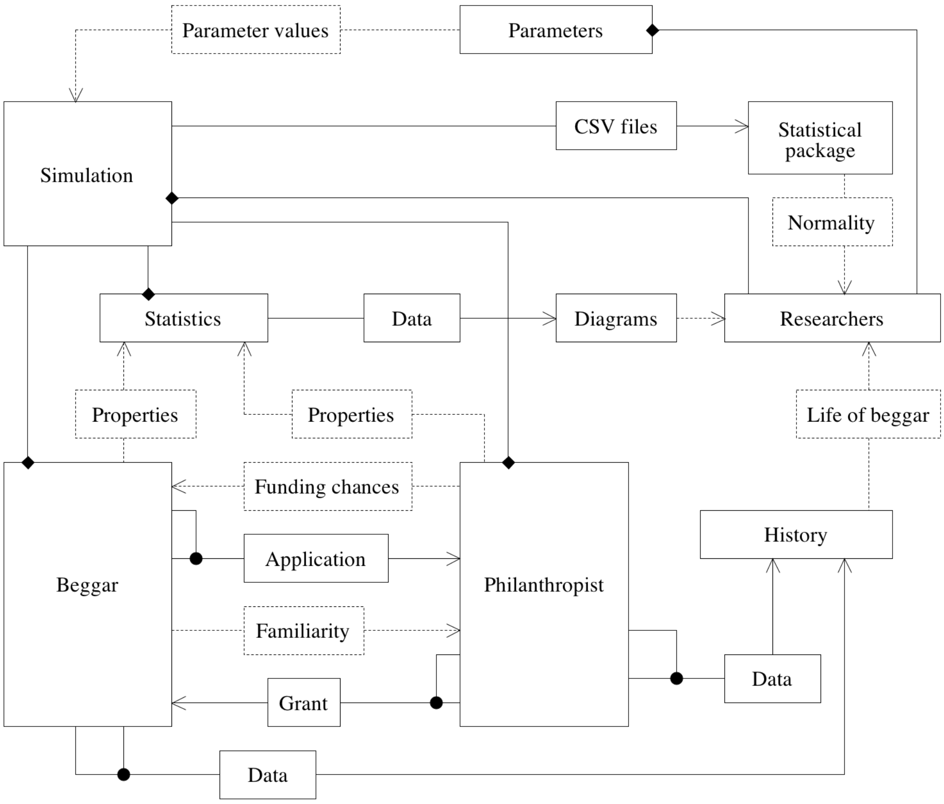

Figure 14 Interaction scenario – Beggars and philanthropists

The beggars and philanthropists can themselves send data about their individual actions to history objects that can be observed by the researchers. The data is recorded when the action happens. One of the purposes of the simulator is to enable the researchers to explain the aggregated behavior by means of the agents' individual actions (histories).

The researchers can control the simulator object: the operation can be run, terminated, suspended, resumed, and stepped through. They can also ask the simulator object to save CSV files to disc from where they can move the files into statistical packages.

6.2 Public library

The following case study was conducted as part of an analysis project at a Danish public library. The purpose of the project was to identify potential improvements to an information search service that the library offers to its users. Briefly, the service is currently executed as follows. A librarian engages in a dialogue with a library user in order to clarify and understand the user's information needs. Based on the obtained understanding, the librarian uses search systems like Internet search engines and reference databases to search for relevant information. The librarian and the user examine the answer set that is returned from the search system. The librarian copies the relevant answers to an unstructured text document (word processor document or email) using cut-and-paste operations. When the search activities are finished, the librarian cleans up the text document and adds relevant comments. The current execution of the information search service has two major disadvantages. First, the answers are handled in a rather low-level manner where the cut-and-paste operations disrupt the interaction between the librarian and the user. Second, the user has to be present at the library and interact physically with the librarian in order to utilize the service.

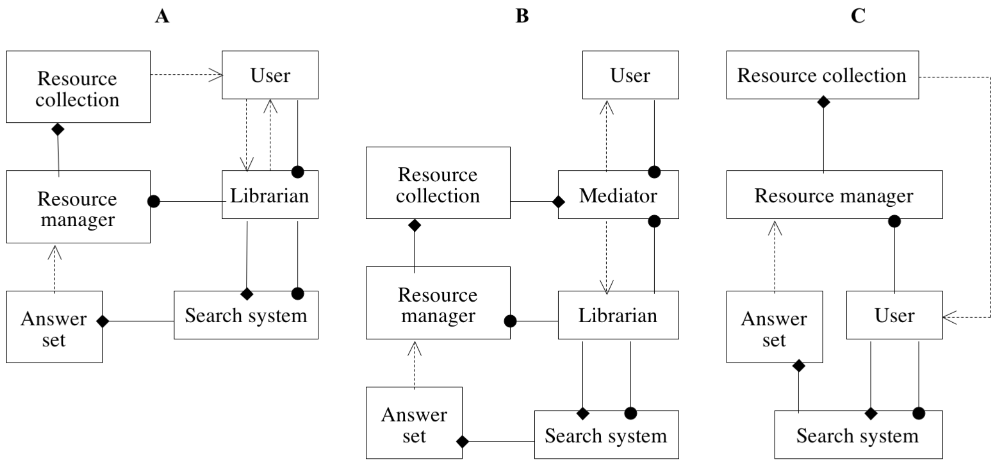

Figure 15 Interaction scenarios – Library

Figure 15 contains three interaction scenarios that represent different views on future search activities. All the scenarios are based on the idea that the handling of answers is supported by a new software component called a resource manager which reads the answer set that is generated by the search system. The resource manager administers a resource collection that contains the current selection of potentially improved answers. The librarian and the user do not interact directly with the answer set and no cut-and-paste operations are necessary. The introduction of the resource manager removes disadvantage number one by eliminating the need for cut-and-paste operations.

In scenario A the librarian handles the resource collection by means of requests to the resource manager. Scenario B is based on the idea that a new software component called a mediator is used to mediate the interaction between the librarian and the user. This eliminates disadvantage number two by facilitating remote interaction between the librarian and the user. In scenario C the user runs the resource manager and the resource collection directly without any interaction with a librarian. The three scenarios can be combined in order to provide the user with a more flexible service.

6.3 Socio-technical activity systems

Interaction scenarios can be used to model socio-technical activity systems in which human beings and technology interact (Mumford 1983; Chae and Poole 2005; Doherty and King 2005; Lune-Reyes, Zhang et al. 2005; Doherty, Coombs et al. 2006). An interaction scenario defines a range of roles that can be played by entities with relevant capabilities.

The interaction scenario in Figure 14 represents a socio-technical activity system in which researchers can use parameters to control a simulator. The simulator in turn controls a beggar and a philanthropist whose behavior can be observed by the researchers. The interaction scenarios in Figure 15 represent different views on information search in a library. Each view represents a socio-technical activity system that allows certain roles to be played by participants with relevant capabilities.

Both the beggars and philanthropists case and the library case represent situations where active and passive entities interact and where the active roles can be played by a combination of human beings and technology. This brings up the question of agency, i.e., the question of whether other entities than human beings can perform actions.

When agency is interpreted in terms of structuration theory, technology cannot be attributed with actions whereas the opposite is true according to actor network theory (Rose, Jones et al. 2005). The notion of hybrids of human beings and technology can be used to overcome some of the problems of agency. The idea is to study the common agency of hybrids rather than isolated human beings and technology (McMaster and Wastell 2005; Ranerup 2007). Our case studies strongly suggest that actions should be attributed to a combination of interaction between human beings and technology.

Socio-technical activity systems can be modeled in many different ways. One modeling approach is action-oriented conceptual modeling, in which the notion of an action (communicative or material) is the basic modeling unit (Ågerfalk and Eriksson 2004). Actions are used to model activity flows and information structures. Activity flows are represented by action diagrams that define actions, their input and output, and their temporal dependencies. Information structures are represented by entity-relationship diagrams in which some entities represent actions and information about actions. Action-oriented architecture (Xiao and Greer 2007), role-activity diagrams (Odeh and Kamm 2003), and action-oriented development (Rittgen 2006) are other examples of action-based modeling techniques.

Interaction scenarios offer a view on action and interaction in socio-technical systems that supplements the dynamic view provided by flow-based approaches and the static view offered by information modeling approaches. For example, the interaction scenario in Figure 14 highlights all the active and passive participants in the beggars and philanthropists case. A flow-based model like a BPMN diagram (White 2004) could be used to highlight the flow of the involved actions and it could represent each active participant by a specific profile. The interaction between the profiles would, however, be limited to message exchanges and the passive participants and their roles in the interactions would not be modeled in a visible manner.

7 Conclusion

We have defined interaction as an activity that involves two or more participants. At least one of the participants must be an agent. This implies that interaction plays two roles in information systems. First, interaction is a source of dynamics that causes an activity system to change. Second, interaction relates the elements of an activity system to each other in a way that supplements logical relations like contracts and functional dependencies. Viewed in this way interaction is a much more fundamental and general concept than its specialized siblings human-computer interaction and human-artifact interaction (Rogers, Sharp et al. 2002).

We have presented and discussed four interaction primitives that play important roles in information systems. For each of these primitives we have discussed potential mediation of the corresponding forms of interaction. An understanding of such mediation is essential if we want to utilize the mediating potential that is an inherent property of information technology. The primitives can be used to characterize interaction within information systems and interaction between information systems and their environments because flows of objects and flows of requests occur both within information systems and between information systems and their environments.

In order to support modeling of all four interaction primitives, it is necessary to combine the two types of flow modeling. Object-based languages subordinate flows of objects to flows of requests. Dataflow diagrams (De Marco 1978) and activity diagrams (Rumbaugh, Jacobson et al. 1999) favor flows of objects. Dataflow diagrams do not support flows of requests. Activity diagrams support flows of requests in a rather limited manner. Event-based languages use events to support flows of requests in a more sophisticated way. Event-activity diagrams utilize shared events with multiple participants to support a general flow of requests where one actor can impose a request on two or more other actors (Bækgaard 2004). Business-oriented diagrams like EPC diagrams (Dehnert 2002; Lübke, Lüecke et al. 2006) and BPMN diagrams (White 2004) are based on the workflow paradigm that primarily focuses on the sequencing of activities within larger business processes.

Future work within this field might include case studies in which interaction scenarios are used to model socio-technical activity systems. Also, it might focus on the formulation of methodological guidelines for the use of interaction scenarios.

8 References

Ågerfalk, P. J. and O. Eriksson (2004). "Action-Oriented Conceptual Modelling." European Journal of Information Systems **13**(1): 80-92.

Alter, S. (1999). "A General, Yet Useful Theory of Information Systems." Communications of the Association for Information Systems **1**(13).

Alter, S. (2006). The Work System Method. Connecting People, Processes and IT for Business Results. Larkspur, CA, Work System Press.

Andersen, P. B. (2006). "Activity-Based Design." European Journal of Information Systems **15**: 9-25.

Bardram, J. E. and C. Bossen (2005). "Mobility Work – The Spatial Dimension of Collaboration at a Hospital." Journal of Computer Supported Cooperative Work **14**(2): 131–160.

Bækgaard, L. (2004). Event-Based Activity Modeling. Conference on Action in Language, Action in Language, Organisations and Information Systems (ALOIS'04). Linköping, Sweden.

Chae, B. and M. S. Poole (2005). "The Surface of Emergence in Systems Development: Agency, Institutions, and Large-Scale Information Systems." European Journal of Information Systems **14**: 19-36.

Checkland, P. (1981). Systems Thinking, Systems Practice, Wiley.

Checkland, P. and S. Holwell (1998). Information, Systems, and Information Systems, Wiley.

Cockburn, A. (2001). Writing Effective Use Cases, Addison-Wesley.

De Marco, T. (1978). Structured Analysis and System Specification, Prentice-Hall.

Dehnert, J. (2002). Making EPCs Fit for Workflow Management. Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten (EPK'02). Trier, Germany.

Denning, P. J. and R. Medina-Mora (1995). "Completing the Loops." Interfaces.

Dietz, J. L. G. (2006). Enterprise Ontology. Theory and Methodology, Springer-Verlag.

Dik, S. C. (1989). The Theory of Functional Grammar. Dodrecht, Foris Publications.

Doherty, N. F., C. R. Coombs, et al. (2006). "A Re-Conceptualization of the Interpretive Flexibility of Information Technologies: Redressing the Balance Between the Social and the Technical." European Journal of Information Systems **15**: 569-582.

Doherty, N. F. and M. King (2005). "From Technical to Socio-Technical Change: Tackling the Human and Organizational Aspects of Systems Development Projects." European Journal of Information Systems **14**: 1-5.

Dunlop, C. E. M. (1990). "Conceptual Dependency as the Language of Thought." Synthese **82**(2): 275-296.

Fillmore, C. H. J. (1968). The Case for Case. Universals in Linguistic Theory. E. Bach and R. T. Harms. London, New York, Sydney, Toronto, Holt, Rinehart and Winston**:** 1-90.

Fillmore, C. H. J. (1977). The Case for Case Responded. Syntax and Semantics: 8. Grammatical relations. P. Cole and G. M. Sadock. New York, Academic Press**:** 59-81.

Goldkuhl, G. (1996). Generic Business Frameworks and Action Modeling. First International Workshop on Communication Modeling. Tilburg, The Netherlands.

Goldkuhl, G. (2001). Communicative vs Material Actions: Instrumentality, Sociality and Comprehensibility. International Working Conference on the Language-Action Perspective on Communication Modelling (LAP'01). Montreal, Canada.

Goldkuhl, G. and M. Lind (2004). The Generics of Business Interaction - Emphasizing Dynamic Features Through the BAT Model. International Working Conference on the Language-Action Perspective on Communication Modelling (LAP´04) New Brunswick, NJ, USA.

Halliday, M. A. K. (1994). An Introduction to Functional Grammar. London, Edward Arnold.

Hevner, A. R., S. T. March, et al. (2004). "Design Science in Information Systems Research." MIS Quarterly **28**(1): 75-105.

Leontiev, A. N. (1978). Activity, Consciousness, and Personality. Englewood Cliffs: NJ, Prentice-Hall.

Lind, M. (1994). "Modeling Goals and Functions of Complex Plants." Journal of Applied Artificial Intelligence **8**: 259-283.

Lübke, D., T. Lüecke, et al. (2006). Using Event-Driven Process Chains for Model-Driven Development of Business Applications. XML Integration and Transformation for Business Process Management (GI-Workshop XML4BPM'06). Passau, Germany.

Luhmann, N. (1984). Soziale Systeme. Frankfurt am Main, Suhrkamp.

Lune-Reyes, L. F., J. Zhang, et al. (2005). "Information Systems Development as Emergent Socio-Technical Change: A Practice Approach." European Journal of Information Systems **14**: 93-105.

March, A. T. and G. F. Smith (1995). "Design and Natural Science Research on Information Technology." Decision Support Systems **15**: 251-266.

Mathiassen, L., A. Munk-Madsen, et al. (2000). Object-Oriented Analysis and Design, Marko.
McMaster, T. and D. Wastell (2005). "The Agency of Hybrids: Overcoming the Symmetrophobic Block." Scandinavian Journal of Information Systems **17**(1): 175-182.
Mumford, E. (1983). Designing Human Systems for New Technology. The ETHICS Method, Manchester Business School.
Nurcan, S., A. Etien, et al. (2005). "A Strategy Driven Business Process Modeling Approach." Business Process Management Journal **11**(6): 628-649.
Odeh, M. and R. Kamm (2003). "Bridging the Gap Between Business Models and System Models." Information and Software Technology **45**(15): 1053-1060.
Ranerup, A. (2007). "Electronic Government as a Combination of Human and Technological Agency: Testing the Principle of Symmetry." Information Polity **12**: 153-167.
Rittgen, P. (2006). "A Language-Mapping Approach to Action-Oriented Development of Information Systems." European Journal of Information Systems **15**: 70-81.
Rogers, Y., H. Sharp, et al. (2002). Interaction Design. Beyond Human-Computer Interaction, John Wiley & Sons, Inc.
Rose, J., M. Jones, et al. (2005). "Socio-Theoretic Accounts of IS: The Problem of Agency." Scandinavian Journal of Information Systems **17**(1): 133-152.
Rumbaugh, J., I. Jacobson, et al. (1999). The Unified Modeling Language Reference Manual, Addison-Wesley.
Russell, S. and P. Norvig (2003). Artificial Intelligence. A Modern Approach New Jersey, Pearson.
Sowa, J. F. (2000). Knowledge Representation: Logical, Philosophical, and Computational Foundations. Pacific Grove, CA, Brooks Cole Publishing Co.
Tesnière, L. (1959). Eléments de syntaxe structurale. Paris, C. Klinksieck.
Vendler, Z. (1967). Linguistics in Philosophy. Ithaca, NY, Cornell University Press.
White, S. A. (2004). Introduction to BPMN, BPMN.ORG.
Winograd, T. and F. Flores (1986). Understanding Computers and Cognition, Addison-Wesley.
Xiao, L. and D. Greer (2007). "Towards an Agent-Oriented Model-Driven Architecture." European Journal of Information Systems **16**: 390-406.

Working Papers from Informatics Research Group

I-2008-04	Lars Bækgaard & Peter Bøgh Andersen: Using Interaction Scenarios to Model Information Systems.
I-2008-03	Lars Bækgaard: Event-Based Conceptual Modeling.
I-2008-02	Sune Dueholm Müller, Lars Mathiassen & Hans Henrik Balshøj: Organizational Change Perspectives on Software Process Improvement.
I-2008-01	Sune Dueholm Müller, Pernille Kræmmergaard, Lars Mathiassen: Managing Cultural Variation in Software Process Improvement: A Comparison of Methods for Subculture Assessment.
I-2006-02	Charles Møller: The Conceptual Framework for Business Process Innovation - Towards a Research Program on Global Supply Chain Intelligence.
I-2006-01	Charles Møller: Business Process Innovation using the Process Innovation Laboratory.
I-2005-03	Charles Møller, Pernille Kræmmergaard & Pall Rikhardsson: The Emergence of Enterprise Systems Management – A Challenge to the IS Curriculum.
I-2005-02	Pernille Kræmmergaard & Jeremy Rose: Managing the ERP implemention journey – change in discourse from classical IT project to technology-driven organisational change initiative.
WP 05-1	Tina Blegind Jensen: Nurses' Perception of an ERP Implementation Process – Based on a Means-End Chain Approach.

Handelshøjskolen i Århus

Aarhus School of Business

ISBN 9788778823113

Department of Business Studies

Aarhus School of Business
University of Aarhus
Fuglesangs Allé 4
DK-8210 Aarhus V - Denmark

Tel. +45 89 48 66 88
Fax +45 86 15 01 88

www.asb.dk