COMPUTATIONAL REPRESENTATIONS OF ACTIVITIES

Peter Bøgh Andersen
Department of Information and Media Studies, University of Aarhus
Helsingforsgade 14, 8200 Aarhus N, Denmark

Keywords: Activities, business processes, mobile computing, context-aware computing, semantic roles, semiotics.

Abstract: Mobile and context-aware technology enables new activity-centred ways of using digital technology that require systematic methods for representing actions and activities computationally. The paper uses findings from ethnography, linguistics and philosophy to paint a generic portrait of activities, and suggests ways of representing it in an object-oriented framework. The paper makes a sharp distinction between the representation and the represented. The representations *are* not activities, they only represent them.

1 INTRODUCTION

This paper discusses the problem of representing activities by means of computational media.

It is motivated by a new emerging use of technology, enabled by mobile and context-aware technology. From a situation centred on documents and applications used from a stationary device, we are moving into a dynamic situation where many documents and services are used intermittently in the same activity, where activities are intertwined, and where use of computers no longer takes place at a certain place, but follows our daily activities as we move around in the world (Bardram 2006). The focus is shifting from documents and applications to the activities in which they are used.

Bardram (2006) suggests the name *activity-based computing* to denote an architecture where activities are the fundamental building blocks and documents and services can be requested as needed from the resources that happen to be present in the environment.

A similar argument is made in Kristensen (2002, 2003). He criticizes object-centric methods and advocates the *association* as a good tool for representing the new kind of computer use. Associations allow dynamic patterns of cooperation and specify the roles its participants may play. The notion of roles recurs in Moran (2005) that sketches an architecture for activity-based software.

Whereas Bardram mainly represents his activity concept through software architecture, Kristensen views associations as a first class citizen that motivates new programming language concepts and explicit formalisms.

In this paper I address the question of how activities can be represented computationally: 1. *what* is to be represented, i.e. what are the characteristics of activities, and 2. *how* may the digital representation look like. To answer the first question, we should listen to ethnographers, sociologists, psychologists, and linguists, whereas the second question belongs to computer science.

Since it is a question of devising signs of something, I shall base myself on semiotics. The representamen of the sign (in the following: representation) is a computational representation that can be executed on a computer, its object is activities in some problem domain, and its interpretant consist in using these representations for conducting everyday activities, guided by rules of interpretation.

Semiotics reminds us that the representation is not identical to its object. Although the two can be related through *similarily* (icons), they can also be held together through *causal relations* (indexes, e.g. via sensors and actuators), and they can in fact be totally unrelated except via *conventions* (symbols).

Computer science has a strong tendency for preferring icons – there must be some kind of similarity between the computational model and its object. In the past, this has lead to an (often criticised) export of good concepts for understanding the behaviour of information systems to also cover an understanding of the object referred to by the systems. For example, since the class-diagrams or state-machines of the Unified Modelling Language are excellent for understanding the structure and behaviour of information systems, they are also assumed to be good

Bøgh Andersen P. (2007).
COMPUTATIONAL REPRESENTATIONS OF ACTIVITIES.
In *Proceedings of the Ninth International Conference on Enterprise Information Systems - ISAS*, pages 95-104
DOI: 10.5220/0002347600950104

for understanding the part of the world represented by the systems.

This has created conflicts between domain experts and computer scientists: domain experts wrinkle their nose at the simplistic nature of the computational representations, whereas computer scientists quickly loose interest in the subtle distinctions of the domain experts.

If we realize that the representation needs not to be similar to the represented, the situation becomes more tractable. There is one set of requirements that should be fulfilled by the computational representation. Apart from being useful for representing their object, given a suitable set of conventions, they must connect to current programming wisdom, since otherwise no system would be produced in the end.

Another set of requirements are valid for descriptions of the object: they must account for what is currently known about the object and preferably in the simplest possible way.

The task is now to make a representation that is in fact programmable and which, given a suitable set of rules of interpretation, will be accepted as representing its object by its users.

Therefore: reproaching the programmer for working with a simplistic view on human activities is just as absurd as reproaching the filmmaker for viewing reality as consisting of cuts, sequences, and scenes, in spite of the fact that the film is accepted by the audience as an exciting thriller. In both cases, the acceptability of the product is filtered through the users' willing suspension of disbelief.

In the rest of the paper, the object – human activities – is mainly described based on linguistic evidence. The reason for this is that language is our main empirical access to the way activities are segmented and classified. When a field-work is started, the events form a confusing flow where patterns and boundaries are hard to see. The only method of bringing order in the confusion is to ask people to describe what they do and why.

If we want to capture the generic recurrent features of activities, the best evidence is the grammaticalized features of language, such as case inflexion, word order, affixes, tenses, etc. The reason is that these features mark distinctions that have been used frequently during a long time. Therefore they probably represent basic distinctions in our understanding of reality.

As for the computational representations, I shall use two popular representations from the UML standard, namely class diagrams and state-machines.

The following preliminary requirements concerning the nature of activities are based on field-work projects in the maritime domain and hospitals (Andersen 2001, 2004a, 2006, Bardram & Bossen 2005, Bødker & Andersen 2005).

Activities must encompass both *material* and *communicative* actions, since they are intermingled in practice. They must deal explicitly with *errors* and failure of equipment, and provide *countermeasures* for such failures. Many activities contain standard countermeasures against past accidents. For example, all pedestrians will look to the right and left for approaching cars before crossing a road (see also Andersen 2001 for a similar analysis of maritime manoeuvres).

Actors must be able to *suspend* and *resume* activities, and *cooperative activities* with several people involved must be possible. Most activities are performed simultaneously with and intertwined with other activities.

Activities must contain *roles*. In both reference domains, there are formal modal hierarchies of roles requiring particular *abilities, knowledge, intentions, rights* and *obligations*. For example, the helmsman must be able to turn the wheel but has no right to determine the course, and certain physical abilities (hearing, sight) are required to work on a ship in the first place. In the hospital, a doctor must be present when resuscitating cardiac arrests, nurses are not allowed to do it alone.

The rest of the paper elaborates these characteristics using findings from linguistics, psychology, and philosophy.

2 ACTIONS AND ACTIVITIES

This section presents a selective overview of research into the structure of actions and activities.

2.1 Roles

2.1.1 The Object of the Computational Representation

Activity theory teaches us (Vygotsky 1962) that human actions are composed of (at least) three different elements that play very different roles: the *subject* intending the action, the *object* towards which the action is directed, and the *mediator* that mediates between subject and object. Linguistics adds a larger number of roles to these three, and connects the role to specific process types (on the relationship between semiotics and activity theory, see Bødker & Andersen 2005).

Roles in linguistics are called *thematic* or *semantic* roles and denote relations between a process and its participants. The theory (e.g. Fillmore 1968, 1977) claims that

- there are a limited number of thematic roles (normally between ten and twenty),
- each process type can be characterized by requiring a small number of obligatory roles, and
- there are important regularities in the way roles are expressed in sentences. Specifically, roles are systematically marked by case-inflexion, prepositions and/or word order.

According to M. A. K. Halliday, the founder of functional-systemic linguistics, the rough rules relating object to representations such as *Birds are flying in the air* are: the process is expressed by means of the verbal group (*are flying*), the participants by nominal groups (*birds*), and circumstances by adverbial groups or prepositional phrases (*in the sky*).

Halliday 1994 presents a list of a dozen process types, each characterized by a specific frame of roles. A more traditional list of roles is given by Jurafsky & Martin 2000. Table 1 shows an adapted version. I have illustrated the table with authentic examples from real life.

The notion of roles is rather common in information systems research. Consider for example the following definition of workpractice:

> A workpractice means that some actors make something in favour of some actors, and sometimes against some actors; this acting is initiated by assignments from some actors, and is performed at some time and place and in some manner, and is based on material, immaterial and financial conditions of transactional and infrastructural character and a workpractice capability which is established and can continuously be changed. *Goldkuhl & Röstlinger 2006: 53.*

Here we meet roles like Agent ('actor'), Beneficiary ('in favor of some actors'), Theme ('something'), Time, Place ('some time and place') and Manner ('in some manner'). Parunak 1995 uses roles as a tool for specifying agents, the ORM database methodology is based on explicit use of roles (Halpin, 1996, 1998), and Sowa 2000 uses it in his conceptual graphs.

In activities as well as and their linguistic representations, there are restrictions as to which participants can fill which roles. For example, abstract concepts liken *sincerity* cannot be the grammatical subject of predicates like "smile" or "have colour", nor is the 2nd officer allowed to plan the voyage of the ship. In fact, the role concept itself implies that the holder of a role has certain rights and obligations and must possess certain abilities.

Table 1: A list of thematic roles. Authentic examples from the maritime domain.

Role	Definition and examples
Agent	The participant initiating and controlling an event: Can *we* berth her without a tug? As *she* goes full speed at shallow water, then *she* creates a water wave
Experiencer	The participant that senses an event: Maybe *we* can see the 'Gudrun' from here.
Theme	The participant most directly affected by an event: Can we berth *her* without a tug?
Material	The participant that changes identity in an event: Isn't that the only place where we get a copy *of those receipts?*
Result	The final identity of the material: We make *a three sixty* (manoeuvre)
Content	An event or a state of affairs: I said to him *that as soon as you were finished steering, you would come down so that we could get it in*
Instrument	The mediator of an event: Can we berth her *without a tug*?
Addressee	The intended recipient of a communicative action: A, have you talked *to the pilot?*
Beneficiary	The participant that benefits from an event: And Sir have you received our pilot chart we have it ready *for you* here
Comitative	The participant playing a role similar to the Agent: Is there a chance that he will come *with the helicopter*, over there, the pilot?
Source	The start location of the Theme of a transfer event: I really thought he came *from Rotterdam*
Destination	The end location of the Theme of a transfer event: 'Gudrun' must sail before we can get *in.*
Purpose	The intention of the Agent of the event: Well, down to about 7.5 meters draught, you need that *in order to run properly with the top of the tunnel.*
Time	The time of the event: he will not sail *until two o'clock.*
Location	The place of the event: we are still lying *here* waiting
Manner	The manner in which the event is performed: Shall we start turning *slowly* now?

In addition, there may be conflicts between these modalities. According to Ryan 1991, activities in narratives must contain conflicts between two or more of the following modalities: *knowledge, intention, obligation*, and *desires*, and between these and the actual world, in order for them to be *tellable*. In melodramas the hero is for example torn between obligations and desires.

But conflicts are also important in non-fiction for diagnosing organisational conflicts: a secretary is *obliged* to finish a report at a certain time, but is *unable* to do so because the information was not delivered to her.

In the technical domain, displays in process control have the important purpose of telling the operator when components are no longer *able* to function in the way they *ought to* according to their specification.

Business processes too can be characterised by the modalities of ability and obligation. In the work-practice definition above, assignments refer to obligations and capability to ability.

Business transactions (Haraldson & Lind 2006) can in fact be analysed as the creation, fulfilment and removal of obligations: the supplier sends a quotation to the customer and they agree on the transaction, meaning that the supplier is obligated to deliver the product and the customer obligated to pay the agreed price. Then the goods are delivered, the customer sign for the product, thereby removing the supplier's obligations, sends the money, and receives a receipt that frees the customer from his obligation to further payments. The *Normbase* system in Liu 2000 represents such processes.

To sum up: there is good empirical reason to posit dynamic bindings between participants and roles consisting of obligations, rights, desires and abilities.

2.1.2 The Computational Representation

The basic computational entity (Figure 1) in the system is simply one that has a name, an identity, a set of sensors and actuators, and belongs to one or more categories. Sensors and actuators are motivated by our emphasis on context aware technology. The categories are necessary, since there are restrictions on who can fill which roles. Entities have two subclasses, *things* and *processes*. Processes are characteristic in being associated with one or more roles that bind other entities to the process. A role is represented by a role-name, plus a binding that contains a number of dimensions representing the participant's modal relations to the activity. A possible representation is depicted in Figure 1. Note that processes as well as things can participate in processes. For example, the participants of the Purpose and Content roles may be processes, as illustrated in Table 1. Note that a participant can be bound to more roles, and that one role may have more participants bound to it. The methods *add* and *remove participant* allow us to manipulate participants during the execution of the activity.

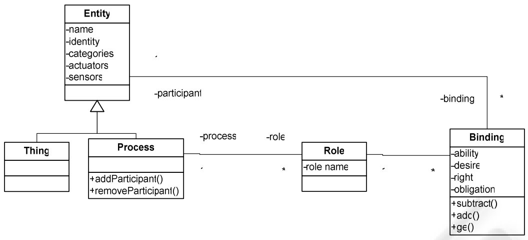

Figure 1: Basic relationships between Things, Processes, Roles and role Bindings.

2.2 Process Types

Process types are useful when we want to create an overview of activities in a certain domain.

Bækgård 2006 proposed five general action types for information systems and in the domain of process control, Lind 1994 proposed a framework, Multilevel Flow Modelling, for modelling mass and energy flows in process plants, and suggested six basic process types involving mass or energy: Sources provide it, transports transport it, barriers block transports, storages store it, balances distribute it, and sinks consume it.

2.2.1 The Object of the Computational Representation

But there is evidence that humans in general systematically distinguish between a limited number of process types. In cognitive linguistics these are called *image schemata* (Talmy 1988, Johnson 1992: 2).

Table 2: Vendler's four verb types.

	Static	Telic	Punctual
Activity	-	-	-
Accomplishment	-	+	-
Achievement	-	+	+
State	+	-	-

Before cognitive linguistics, Zeno Vendler (1957, 1967) classified verb meanings according to the dynamic structure of their referents. According to him, there are four major types of processes, namely *Activities* (play football – Vendler's activity should not

Table 3: Four basic control processes. T = change, d = doing, acting. Condition: what happens if no action is taken.

Condition	Explanation	Action	Explanation	Result of action	Explanation
pT¬p	p exists but vanishes unless maintained	d(pTp)	p is maintained	pTp	p remains (duratives)
¬pT¬p	p does not exist and does not happen unless produced	d(¬pT p)	p is produced	¬pTp	p happens (ingressives)
pTp	p exists and remains unless destroyed	d(pT¬p)	p is destroyed	pT¬p	p vanishes (cessatives)
¬pTp	p does not exist but happens unless suppressed	d(¬pT¬p)	p is suppressed	¬pT¬p	p remains absent

be confused with activity in activity theory), *Accomplishments* (drive to the parking lot), *Achievements* (win a race; in computer science we talk about state changes) and *States* (sleep, be a plumber). The processes differ in the terms of three oppositions: static/dynamic, telic/non-telic, and punctual/non-punctual (Van Valin & LaPolla, 1997), as shown in Table 2.

In activities an action is repeated an indefinite number of times. In accomplishments there is also action but it stops when a certain limit has been reached, e.g. when I have arrived at the parking lot. Achievements denote a momentary state-change, and state terms denote the continuation of a state of affairs. Vendler proves the linguistic existence of these types by showing that grammatical features, such as ing-forms and adverbials of time and duration, and, I may add, in Scandinavian languages, the auxiliary of the past participle, depend upon them.

In fact, the observation is much older; the idea that language structures processes into a few types, was described more than a hundred years ago under the heading of *Aktionsart* (e.g. Noreen 1903-1923). Suffixes and prefixes are frequent markers of Aktionsart: *-en* as in *blacken* (ingressives), *-de* as in *decolourize* (cessatives), as well as aspectual verbs like *begin, keep, stop* (*keep running,* durative).

Lind (1994) proposes an analysis of basic processes in industrial process control that is reminiscent of the old Aktionsart-theory, but was in fact inspired by Von Wright's behavior analysis for defining four basic kinds of actions (Table 3).

2.2.2 The Computational Representation

In the following Vendler's analysis is used to describe the dynamic morphology of processes, whereas Lind's classification is used to classify the effect of one action on another. Processes are divided into four subprocesses representing Vendler's four classes (Figure 2).

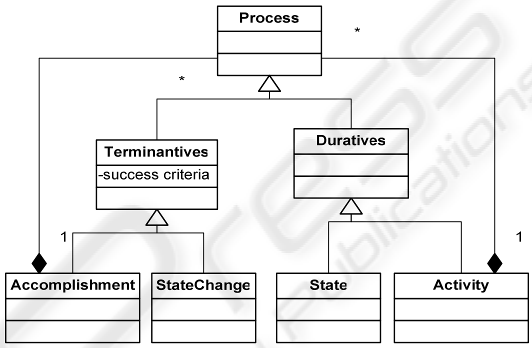

Figure 2: Representation of Vendler's four process types.

The class of terminatives – accomplishments and achievements/state-changes – share the feature of having success criteria that define when the process has ended with success. Accomplishments and activities have other processes as components (Figure 2). All classes can be in an active or suspended state (Figure 3), and each class is differentiated by the state-changes it undergoes inside its active state.

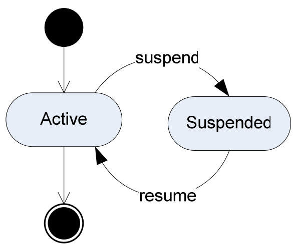

Figure 3: The shared structure of all processes.

Accomplishments and states are exemplified in Figures 4 and 5.

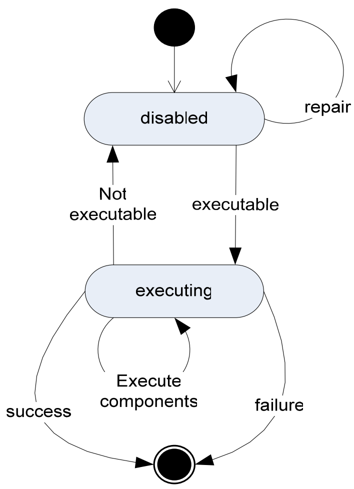

Figure 4: The internal structure of the active state of accomplishments.

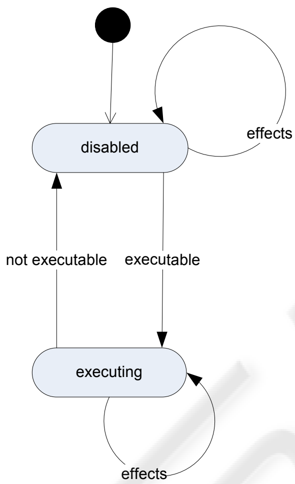

Figure 5: The internal structure of the active state of states.

Both can shift between a disabled state and an executing state when they receive the messages *executable/not executable* from their participants (Section 3). For example, the 10 cm radar announces to the officer that it has become less able to participate in steering the ship; the steering activity is therefore not optimally executable before the 10 cm radar is replaced by its 4 cm colleague. Thus, participants can desert from and become enrolled in the activity at execution time as part of normal operating procedures.

The accomplishment is meant to represent a goal-directed action performed by an Agent. When the Agent executes an accomplishment, he executes the processes represented by its components; the process ends if it receives the messages *success* or *failure* defined by the success criteria. If the process is in the disabled state, the Agent can try to find remedies for repairing the defects.

A state is only defined by the effects it has on other processes, both in its executing and disabled state. It has no defined ending point, and does not contain component actions. Examples: a patient may be conscious which enables a certain set of behaviors (talking, eating, etc.), or he may be in a coma, which disables these behaviors. A ship may be moving, which enables steering, or it may lie still, which disables steering.

Vendler's activity is represented as a state with component actions added. For example, a running pump may be driven by a combustion engine that performs an activity where pistons move back and forth regularly. But note that we are talking about representations: a real process may be represented by a state or an activity, depending upon the aspects we are focusing on. If we disregard the movements of the piston and are only interesting in the flow produced by the pump, we would represent it as a state, not an activity.

Vendler's achievement is an accomplishment with component processes removed. Whereas accomplishments have duration – they repeatedly perform their component actions – achievements are momentary.

3 EFFECTS AND FAILURES

3.1 The Object of the Computational Representation

We now need to specify what failure and effect means. We propose the following definitions:

An action executes if and only if.

1. All participants are able to fulfil their roles to some degree.
2. The Agent filler is strongly obligated and/or strongly desires to fulfil his role.

This means that an action is disabled if one or more participants are no longer able to fulfil their roles, and/or the Agent is no longer obligated or no longer desires to fulfil his role. In accordance with these definitions, the effect of actions are defined as

3. a change of the participants' role-bindings to other actions, possibly to newly generated actions.

Note the phrase *to some degree* in (1). It is not required that all participants perform faultlessly, since in this case, we would be blind to the parts of reality that is full of degraded pumps, corroded pipelines, inattentive operators, and worn tools. This would be inappropriate if we used the theory to design MIMIC diagrams for operators of process plants, since a

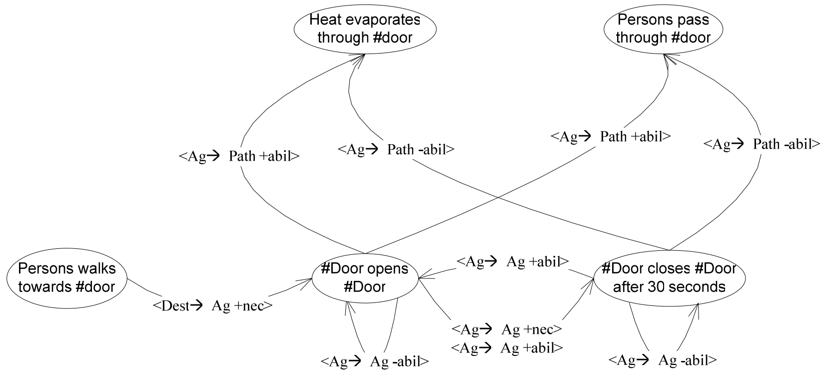

Figure 6: The door-opener as part of an activity system. Dest = destination. Ag = agent. Path = path.

Table 4: Four types of effects.

Process	Definition
D maintains P	P is executing and D further increases the role-bindings between an able participant and P. Example: D = the captain turns the wheel, P = the captain steers the course by means of the rudder. The rudder's ability to participate in keeping the course is increased to counteract the effect of wide-wind.
D produces P	P is disabled and D increases the role-bindings between a defect participant and P. Example: D = the chief starts the engine, P = the engine produces rotational energy. The engine's ability to function as the Agent in energy production is increased.
D destroys P	P is executing and D decreases the role-bindings between an able participant and P. Example: D = turning off the fuel supply, P = an engine is running by means of fuel. The fuel is disabled to function as an instrument for the process.
D suppresses P	P is disabled and D further decreases the role-bindings between a defect participant and P. Example: D = a cooler moves water through the walls of a burner, P = the burner melts its walls. The walls' ability to participate in the melting process is further decreased.

main purpose of such diagrams is to alert the operator of suboptimal equipment.

In the definitions so far, there is nothing hindering machines in participating as Agents in activities, quite in line with Actor Network Theory. This does not imply that humans and non-humans are equally qualified as Agents in all activities. A clock is perfectly suited for being an Agent in the physical process *The clock ticks* (which is also the way language treats it), but is absolutely unqualified to participate in the business process *the clock sends a quote to its customer*. As ANT rightly claims, agency is a function relating a participant to an network, not an inherent property of a participant.

3.1.1 Effects

Lind's four process control types can now be redefined as shown in Table 4.

The difference between these material actions and communicative actions is that material actions influence the ability of participants, whereas communicative actions influence their desires, rights and obligations.

Thus, *A requesting B to do C* presupposes that B would not have done it by himself, and has the intended effect that B assumes an obligation to do C (cf. Searle 1994: 65 ff: A attempts to get B to do C); in other words: the request has the intended effect to increase the role-binding associating A to the role of Agent in the action C. It is thus an example of *produce*, whereas the act of reminding B to do C is a case of *maintenance*. Communicatively *destroying* something is exemplified by a moderator removing a topic from the agenda, and *suppressing* is to prevent embarrassing topics from popping up.

Since communicative actions involve abilities, instrumental actions can influence communicative ones. For example, a VHF-radio is not able to participate in the communication "The captain ordered the first officer to let the lines go on the VHF", unless the captain is close to the VHF. This can be achieved by the captain taking hold of it. Conversely, the first officer is not entitled to participate in the instrumental action of letting the lines go before ordered by the captain.

Above I have assumed that machines as well as humans can fill the Agent role; taken literally this means that to switch on the ignition is to obligate the machine or make it strongly desire to run. This is clearly not a good description of how machines behave. Here is a better solution.

The point of departure is that all modal dimensions are organized in a weak and a strong operator. In classical modal logic, they are called N = necessity and M = Möglichkeit. They are related as shown in (4).

$$N\,P = \neg\, M\, \neg P \qquad (4)$$

When we describe human actions, we need to distinguish between deontic (obligations) and axiological (desires) variants.

$$\text{obligated } P = \neg \text{ allowed } \neg P \qquad (5)$$

$$\text{desire } P = \neg \text{ inclined } \neg P \qquad (6)$$

The reason is that for humans to be obligated to do something can be quite different from desiring it, so two different modalities are clearly necessary here – an inner and an outer necessity that may conflict.

However, for machines, the two are merged into one, so here ‘desires’ and ‘is obliged to’ are synonymous with ‘is forced to’ (7). The difference between human and machine agents is thus that humans are bound to activities by a variety of modal ties which coalesce into one in machines.

$$\text{forced } P = \neg\text{able } \neg P \qquad (7)$$

(7) can be found in descriptions of business processes where a process can *trigger* another process (force it to execute) or it can *enable* it (Rittgen 2006).

In the following, I shall only use necessity (forced to) and possibility (able to) where machines are concerned.

The effects of action A on action B can be diagrammed as shown in Figure 6 that describes an automatic door-opener. We draw an arrow from A to B and annotate it with the modal change <*Role*$_a$ → *Role*$_b$*, change, dimension*>, meaning that if someone has participated successfully in action A as Role$_a$, his likelihood to participate in action B as Role$_b$ is changed along the specified dimension. For example, when a person walks towards the door, the door participates as Destination in the action; this forces it to participate as Agent in the opening action. When it opens, it looses the ability to participate in opening, but is enabled to participate in the action ‘heat evaporates through the door’ and ‘persons pass through the door’ in the role of Path.

Note that the actions are half-baked, as is the case with most actions (Andersen 2004a, 2006). The *#door* participant (# symbolizes an instantiated participant) is filled in and fixed, but the person participant is only specified by its category. If a car tried to participate, it would fail because it is of the wrong category. The idea is that we seldom construct activities from scratch because the environment already offers us half-baked actions to use.

Note also that the behaviour of the door-opener is formulated as part of a network of actions involving humans and machines. The basic construct is an activity comprising humans and machines, and the machine is described as a participant in this activity (see Bødker & Andersen 2005 for maritime examples of this analysis). This makes it possible to generate understandable descriptions of its behaviour since it is described in terms of activities humans participate in. In the concrete case of the door opener, understandability presents no problem, but ease of verbalization and visualization becomes an advantage if we are faced with complicated dangerous machinery, such as nuclear power plants where particularly abnormal situations present a problem.

3.1.2 A Classification of Modal Changes

Some types of modal changes are very frequent and represent phenomena known from the literature.

Theme → Instrument. Suppose that my lawn-mower fails to fill the Instrument role in *I am mowing the lawn with my lawn-mower.* Then my focus shifts from my lawn to my mower (on focus shifts, see Bødker 1996), *I repair the mower*: <*Instr → Theme*>. When the mower is fixed, the reverse process happens and the instrument’s ability to participate in *mowing* is increased, <*Theme → Instr +abil*>.

Destination → Source. Similarly, if I know there is a flight connection from Copenhagen to London, and I myself am located in Stockholm, then travelling from Stockholm to Copenhagen will enable me to later catch the plan to London. In this case, the Destination of the subordinate action is identical to the Source of the super-ordinate action, and participation in the subordinate action increases the airport’s ability to participate in the super-ordinate one: <*Dest → Source +abil*>. This modal change underlies flow-diagrams, such as the ones in Lind 1994.

Instrument → Agent. A third example of role-shifts is in automatic systems. They often consist of components with responsibilities that can be described by means of roles. A higher-level component uses a lower-level component as Instrument and delegates certain tasks to it in which it is forced to act as the Agent: <*Instr → Agent +nec*>. For example, the autopilot maintains the course by means of the rudder servo system and delegates the task of maintaining the angle of the rudder to the servo system (Bødker & Andersen 2005).

Addressee → Agent. Transfer of control where *A orders B to do C* causes *B does C* to execute can be described as the role-shift <*Addressee → Agent +obl* > where the addressee of the first action is obligated to become the Agent of the second one. Bækgård 2006 suggests that such control patterns should be among the basic building blocks in information systems.

In addition to modal changes, we also regularly meet participant changes. For example, *Automation → Operator,* where automation replaces a human operator in the role af Agent.

3.2 The Computational Representation

Digitally, the schema <$Role_a \rightarrow Role_b$, *change, dimension*> relating action A and B can be described by an Effect object that specifies the modal changes and is associated to the two role objects, $Role_a$ and $Role_b$, that are in turn associated to A and B (cf. Figure 1).

In principle, all participants get qualified by participating in other processes, but what about time and place? The action *I board the train at the platform now* requires that the train and I are located at the platform of course, but the action can still not execute at 4 pm if the timetable says 5 pm. Time can be treated just as the other participants, since the *now* too participates in a process, namely the passing of the time, which is measured by the clock sensor. Only when the clock points to 5 pm, is the *now* able to participate in the train departure.

In many activities, participants are only qualified if they are assembled at the same location. This can be measured by location aware technology.

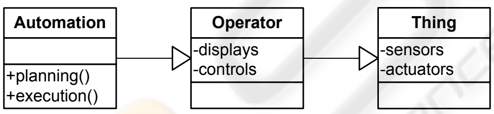

Figure 7: Classes representing the human operator and an automatic system.

The shift *Automation → Operator* can be supported computationally as shown in Figure 7. We make the Operator a subclass of Thing that inherits sensors and actuators. The Operator class is for manual operation, so it adds displays and controls that let humans use sensors and actuators. Automation is a subclass of Operator and inherits the displays and controls, but adds methods for planning and execution. The switch *automation → operator* can now be represented as exchanges of instances of the Automation class and the Operator class. The benefits of letting Automation inherit the displays and controls of the Operator class is that the human can see what the automation is doing, and is able to override the automation by using the controls. The former allows the operator to understand what the automatic system does (Bødker & Andersen 2005) and the latter is used in cars and ships with automatic cruise control to allow manual intervention for safety reasons.

4 SUMMARY

Motivated by new activity-centric uses of IT, I have drawn a generic sketch of activities and suggested ways of representing it computationally. An activity is associated to one or more roles to which participants are bound by modal ties of various strengths. Activities can be classified in a few morphological types. They execute when all participants are able to fulfil their roles and the Agent is sufficiently obligated or motivated. The effect of executing an activity is to change the modal ties of its participants to other activities. During execution participants can desert and new participants enrol.

REFERENCES

Andersen, P. Bøgh, 2001. Maritime Work and Communication. *Australian Journal of Information Systems*, 8(2): 83-102

Andersen, P. Bøgh, 2004. Anticipated Activities in Maritime Work, Process Control, and Business Processes. In K. Liu (ed.), *Virtual, Distributed and Flexible Organizations. Studies in Organizational Semiotics*. Dordrecht: Kluwer Academic Publishers, 35-60.

Andersen, P. Bøgh. 2006. Activity based design. *European Journal of Information Systems* 15(2), 9–25

Bækgård, L., 2006. Interaction in Information Systems – beyond human computer interaction. *Proceedings of ALOIS 2006.* 257-272. Borås, Sweden: University college of Borås. http://www.vits.org/?pageId=317.

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

Bardram, J. E., 2006. From Desktop Task Management to Ubiquitous Activity-Based Computing. In Victor Kaptelinin and Mary Czerwinski (eds.), *Integrated Digital Work Environments: Beyond the Desktop Metaphor.* Cambr., Mass: MIT Press.

Bødker, S. & P. Bøgh Andersen, 2005. Complex Mediation. *Human-Computer Interaction*. (20), 353–402.

Bødker, S., 1996. Understanding computer applications in use. In: Holmquist et al., (eds), *Signs of Work.* Berlin: de Gruyter, 325 -348.

Fillmore, Ch. J., 1968. The case for case. In: E. Bach & R.T. Harms (eds.), *Universals in Linguistic Theory.*

London, New York, Sydney, Toronto: Holt, Rinehart and Winston, 1-90.

Fillmore, Ch. J., 1977. The case for case reopened. In P. Cole and G. M. Sadock (eds.), *Syntax and Semantics: 8. Grammatical Relations*. New York: Academic Press, 59-81.

Goldkuhl, G. & A. Röstlinger, 2006. Context in focus: transaction and infrastructure in workpractices. *Proceedings of ALOIS 2006*, 41-57. University college of Borås: Borås, Sweden. http://www.vits.org/?pageId=317.

Halliday, M. A. K., 1994. *An introduction to functional grammar*. London: Edward Arnold.

Halpin, T., 1996. Business Rules and Object Role Modeling. *Database Programming & Design*, vol. 9, no. 10, 66-72. Retrieved March 1 2007 from http://www.orm.net/ overview.html.

Halpin, T., 1998. Object Role Modeling (ORM/NIAM). In: Bernus, P., Mertins, K. & Schmidt, G. (eds.), *Handbook on Architectures of Information Systems*, Springer. Retrieved March 1 2007 from http://www.orm.net/overview.html

Johnson, M., 1992. *The body in the mind*. Chicago and London: The University of Chicago Press.

Jurafsky, D. & J. H. Martin, 2000. *Speech and Language Processing*. New Jersey: Prentice-Hall.

Kristensen, B. B., 2002. Associative Modeling and Programming. *Proceedings of the 8th International Conference on Object-Oriented Information Systems (OOIS'2002),* Montpellier, France, 2002

Kristensen, B. B., 2003. Associations: abstractions over collaboration. *Proceedings of the 2003 IEEE International Conference on Systems, Man & Cybernetics,* Washington, D.C., USA, 2003.

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

Liu, K., 2000. *Semiotics in Information Systems Engineering*. Cambridge: Cambridge University Press.

Moran, T., 2005. Unified Activity Management: Explicitly Representing Activity in Work-Support Systems. *ECSCW 2005 (European Conference on Computer Supported Cooperative Work) Workshop on Activity: From a Theoretical to a Computational Construct*. Paris, September 20, 2005

Noreen, A, 1903-1923. *Vårt Språk,* vol. 1-7. Lund: Gleerup.

Parunak, H. Van Dyke, 1995. Case grammar: a linguistic tool for engineering agent-based systems. Retreived March 1 2007 from http://www.newvectors.net/staff/parunakv/casegram.pdf

Rittgen, P. 2006. Towards a language for business action theory. *Proceedings of ALOIS 2006*, 3-16. University college of Borås: Borås, Sweden. http://www.vits.org/?pageId=317.

Ryan, M-L., 1991. *Possible Worlds, Artificial Intelligence and Narrative Theory*. Bloomington & Indianapolis: Indiana University Press.

Searle, J., 1994. *Speech Acts*. Cambridge: Cambridge University Press.

Sowa, J. F., 2000. *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Pacific Grove, CA: Brooks Cole Publishing Co.

Valin, Van R. D. & R. J. Lapolla, 1997. *Syntax*. Cambridge: Cambridge University Press.

Vendler, Z., 1957. Verbs and times. *The Philosophical Review*. 143-160.

Vendler, Z., 1967. *Linguistics in Philosophy*. Ithaca, N.Y: Cornell University Press.

Vygotsky, L.S., 1962. *Thought and Language*, Cambridge, MA: The MIT Press.