CONTEXTUAL ACTIVITY MODELLING OF INFORMATION SYSTEMS

Göran Goldkuhl
Dept of Computer and Information Science
Linköping University
S-581 83 Linköping
Sweden
tel + 46 13 281452
fax + 46 13 142231

ABSTRACT

Functional modelling of information systems (IS) is often top down, like Structured Analysis and ISAC. Dynamic aspects of IS and activities are not fully described. These approaches have an IS-centered view. When describing IS in relation to surrounding activities the IS is described as a unified whole. The paper presents an alternative approach to modelling IS and activities: Contextual activity modelling with the description technique Action Diagrams. This is a graphical notation. It is possible to describe information flow as well as material flow. The dynamics of the activity pattern is described in terms of action logic. The description of IS is contextualized. IS is described as several different actions and each action appears in its "own" context. Hierarchical decomposition is rejected as a main descriptive and analysis approach. Instead a contextual approach is preferred. Different activity contexts are described and linked together.

1. INTRODUCTION

There are different approaches to modelling of information systems (IS). Information engineering approach is sometimes contrasted with functional modelling. The information engineering approach puts emphasis on conceptual modelling of information. One main assumption is that modelling of information should be done independent of application and usage. This approach is aiming at a stable view of information. Information engineering view has been theoretically challenged [11, 22] for being epistemological naive. In practice functional modelling has probably been the prevalent approach. Later these two approaches seem to have merged. Functional approaches - like Structured Analysis [30] - have often added and integrated some conceptual modelling technique.
These two approaches focus different aspects of IS: Information systems as
- conceptualization and
- action.

Both of these aspects are necessary to model [23]. No one can be left out if you want to
make a description comprehensive enough as a basis for design, implementation and usage.

The need for modelling these two aspects can be explicated by putting a communication
perspective on IS. Every information system involves communication to and from users.
Communication is not only a transmittance of information. It also involves other linguistic
functions. Communication involves expressability and intentionality. Through
communication social relations are established between sender and receiver. This is well
described in speech act theory [24, 15]. In communication there is a propositional/conceptual
aspect and a social action aspect. When taking a communication perspective on IS, one must
model both conceptualization and action of IS.

This paper follows the emerging tradition of language action view on IS [1, 8, 9, 11, 12,
22, 28].

The paper presents a modelling method taking an action emphasis. I will contrast this
approach to some traditional methods for functional modelling; what is new, what is
different, which are the common principles?

Taking an action perspective means that such aspects come in foreground in the models (and
the conceptual in the background). There must be other related and supplementary models
with conceptual aspects in foreground.

2. A CRITICAL INVESTIGATION OF TRADITIONAL METHODS FOR
INFORMATION SYSTEM MODELLING

2.1 Two top down methods for functional modelling

Action/functional modelling often means that the relations between IS and its user/activity
environment are modelled. There are several methods for analysis and description of
information systems and their relations to surrounding organizational activities. I will
mention two well-known examples of methods: Structured Analysis (SA) with Data flow
diagramming [7, 30] and the ISAC approach with Activity graphs [20, 21]. Structured
Analysis seems to be one of the most used methods especially in North America. The ISAC
approach is known and used especially in Scandinavia and The Netherlands. I will use these
two well-known approaches as reference and comparison for my analysis and presentation.
There are of course other approaches [e.g. 6, 18, 27] in the area of IS modelling putting focus on other aspects within a dynamic framework.

There are differences between these two methods but there are also certain similarities. Both methods are working with
- graphical notation and
- hierarchical decomposition.

Graphical notation is used in order to elicit and visualize complex structural relationships. Both methods emphasize the graphical notation as a means for enhancing user - analyst communication. There are rather simple notational rules. Data flow diagrams and Activity graphs are used as essential models in respective methodological context. There are other methods and models which they relate to; i.e. they should be used together with these other models.

Hierarchical decomposition is both an analysis and description approach. The rationale behind using hierarchical levels of description is that
- different sub descriptions can be related to each other in a controllable fashion and that
- both overview and detail can be provided.

There are differences between the methods concerning drawing rules and symbol forms. The layout of the graphs will be different. This can be considered as a matter of taste. People will like the notation which they are accustomed to. Probably the layout quality is not however only a matter of taste and habit. There might be important differences in easiness to interpret and overview.

There is another important difference between SA and ISAC; the scope of description. Data flow diagrams describe the information flow and also external entities (actors and other computer-based systems). In Activity graphs (of ISAC) both information flow and material flow are described together with different sub activities. The ISAC approach has a broader description scope. This is the main reason why ISAC should be preferred to SA.

2.2 A critical look: Implicit properties and some experiences

So far this description has focused on principles which are often explicitly mentioned in methodological texts [21, 30]. There are also important implicit properties.

The two methods have a similar approach in describing information systems.

An information system is seen as a unified part.
The methods contain a IS-centered view. This IS-centered view together with hierarchical decomposition makes it difficult to model and understand dynamic aspects of the organizational activities and the role of the information system as a part within.

Working with top down partitioning is not unproblematic. The author has more than 12 years experience of active use of the ISAC method. Some important observations from using such graphs are: Hierarchical decomposition gives rise to many changes in graphs. This is mainly due to balancing the different levels of description. Following strictly the top down approach will give intermediate graphs of minor interest. The interesting information is usually on the top and bottom (detail) levels. Also due to the top down principle there will be graphs which are cluttered (compact) and others which consist of little description value. The describing analyst is forced by the hierarchical structure and is therefore restricted in what to select and describe in a certain graph.

There are important aspects concerning the interplay between surrounding activities and IS which are not modelled in ISAC or SA graphs. Examples of this are triggers, sequentiation and alternatives. This is more thoroughly described in section 3.2 below. The kind of IS modelling performed in SA and ISAC makes it hard to understand the dynamic behaviour of IS in relation the organizational activities. The difficulties to model dynamics in Data flow diagrams have been identified [3, 5]. There are attempts to improve Data flow diagrams with dynamic properties, like e.g. triggers and selections [ibid]. The improvements are probably only possible to a certain level. Beyond that level there must be a total reconstruction of the logic of Data flow diagrams, i.e. a totally different modelling technique.

2.3 Some conclusions

There are positive experiences using these methods. Their popularity should probably be found in their
- abilities to model important aspects, and
- communication strengths through simple graphical notation.

The graphical notation makes it possible to investigate and to visualize complex structural circumstances. Graphical notation in itself does not provide enough analysis support. The semantics of the notation must be sufficient in describing important aspects of IS in relation to organizational activities. The dynamics of IS seem to be important to model.
I summarize conclusions from this analysis of the two traditional methods:

- Graphical notation is important but not sufficient
- A need to overcome the anomalies in hierarchical decomposition
- Questioning of describing IS as a unified whole
- A need for better description of the dynamics of IS in relation to organizational activities.

3. CONTEXTUAL ACTIVITY MODELLING - SOME UNDERLYING PRINCIPLES

This paper presents an alternative method for analysis and description of IS in relation to organizational activities. This approach is called Contextual activity modelling. The main notation is Action Diagram.

3.1 IS is an integrated part of an activity pattern

Every method for IS modelling builds on (implicit or explicit) ontological assumptions concerning information system. Contextual activity modelling builds on an action and communication perspective [9, 11, 12]. Contextual activity modelling emphasizes the information system as part of an activity pattern. IS should give support to users performing different tasks in the organizational activities. One main principle behind Contextual activity modelling is:

*Computer-based and manual functions should together form a coherent and congruent pattern of activities.*

How this activity pattern is built up will have decisive consequences for organizational efficiency.

What do I more precisely mean by an activity pattern? What are the basic building blocks of an activity and an activity pattern? What ontological assumptions are behind this approach?

An activity consists of actors, actions and objects. There are human actors in specific activity roles (e.g. warehouse foreman and warehouse worker). These actors are performing actions. There are certain organisational tasks like receiving, unpacking and storing goods. Actors use resources and instruments (e.g. special control equipment) in their actions. They act upon objects, i.e. objects are used as basis in their action. These objects can be transformed in action. Action has results (produced objects), e.g. unpacked goods, stored goods. Action objects can be material objects or information (messages). Information produced in action means communication.
Some actions can be performed with the aid of automatons, e.g. computers with software. In such cases a computer-based information system performs operationally an action based on human specified rules and instructions for this action.

A note on the use of the *action* concept in this context: When I talk about IS action I emphasize that when people use IS they instantiate action. I am not taking a reifying standpoint saying that the information system acts intentionally by itself. IS means rule-governed action and these rules always emanates from intentional human beings.

An activity pattern means that different actions within an activity are related to each other in certain ways, e.g. a shipment of goods must be preceded by an order of these goods. The relations between different actions build upon such social action logic of the activity. How this pattern is structured depends on goals, rules and knowledge within this activity.

Another way to describe activity pattern is by using the concept of *action game*. I have borrowed and expanded the concept of *language game* from the later Wittgenstein [29]. An action game involves a language game, but it can also include instrumental material action, like e.g. unpacking goods. An action game is an institutionalized pattern of different language and material actions; with rules both for single acts and how different acts are combined and interrelated [8, 9, 11]. Using the word game I emphasize the *rule-boundness* and *dynamic* character of action. An IS is considered to be part of an action game in the organization. An action game has rules and structure, and its character is dynamic.

IS in action is considered to consist of rule-based functions. The functions of an IS involve handling of information and communication. There are rules both for IS action as such (rules for inference, for information contents and structure and for communication) and how these actions relate to other parts in the activity pattern (action game) which the system is part of.

An activity - in this meaning - consists of action ("processes") and action objects (information and material objects). The semantics of Action Diagrams makes possible descriptions of both communication ("information flow") and handling of material objects ("material flow").

### 3.2 Action logic

Contextual activity modelling means designing and modelling this activity pattern and its different IS functions which are part of it. The Action Diagrams should model different parts of activity patterns. This includes modelling of "information flow" as in other methods. But it includes also other important aspects. It must be possible to model the dynamic character of both IS and its surroundings activities. This is called *Action Logic*. 
Action Logic means modelling of

- sequential order of actions
- alternative actions (decision points)
- conjunctive actions
- contingent actions (i.e. actions occurring only sometimes)
- trigger of actions (by time or communication)
- interruption of actions (by time or communication)
- conditions for actions
- parallel actions

3.3 Contextualization of IS action

The IS functions are seen as actions in the activity pattern. When using Action Diagrams the IS is not described as a unified whole. The IS is described as several functions and in several places according to dynamics of the activities. The action character of the IS is emphasized. The different IS functions (actions) are descriptively contextualized. This means that each IS function are described in the specific activity context where it shall "take place".

This contextualization of IS is one of the most important aspects in this approach. The traditional way, as described in section 2.2 above, is to consider IS as a whole in relation to the surrounding environment. In Action Diagrams the information system is described in several different (but related) activity contexts.

3.4 Instead of top down: Contextual descriptions

Top down analysis is very often taken for granted in many methods. Top down is a prevalent "myth" in the IS area as Hirschheim and Newman state it [16]. And as such it is hard to challenge. It has been implemented in many methods, as e.g. the two methods mentioned above: SA and ISAC [21, 30]. It has also been given theoretical explication [19]. There are however difficulties applying a top down approach. This was mentioned in section 2.2 above.

The analysis approach taken in Contextual activity modelling is not hierarchical decomposition. The approach is best described as contextual. Different activity contexts are described. One can use a decomposition approach but it is not at all needed. One is not limited by an hierarchical description structure. The modelor chooses by himself the most appropriate description structure. The inherent structure of the Action Diagram modelling technique gives support to choose a contextual description structure. An activity context can be described in one page and then related (by simple connectors) to other activity contexts.
of the total described activity pattern. Action diagrams are contextualized models. Contextual activity modelling is more a "bottom all" approach than a "top down" approach.

4. ACTION DIAGRAMS

4.1 Basic elements

Action Diagrams are graphical models. They have a well defined notation. They are intended to be used by systems analysts and IS users together in specifying and modelling information systems. The modelling aproach tries to combine different qualities of descriptions. They should be both precise and easy to read and understand. An intelligible graphical notation is prefered instead of too sophisticated formalisms. Semantic expressibility of dynamic aspects of IS and organizational activities is the key idea behind this approach. The reason for this is that the IS must function in a dynamic environment. Modelling the dynamics of that environment and the action role of the IS within it, is supposed to produce IS of better quality. Such IS are more adapted and supportive to and integrated in the surrounding organizational activities.

The central part is description of action and action objects (i.e. information or material objects). The semantics of describing actions are important. It consists of describing WHAT is performed (the task) and WHO is performing. WHO - the doer - can be human actors or computer-based information systems or some other mechanical instrument. In figure 1 the main descriptive elements are expressed. A more thorough description of the notational rules is found in the appendix. The complete description of the notational rules is [10].

![figure 1 The main descriptive elements of Action diagrams](image-url)
4.2 An example

A simple example with two Action Diagrams is used below (figure 2 and 3).

A short description of the example: *Spare parts supply*. The example concerns an organizational unit dealing with supply, storage and delivery of spare parts.

Development of the actual activity is aimed at better control of the spare parts supply and to shorten the lead times from reception to delivery. The development includes an improvement of the information system for spare parts supply control (SPSC). One important change is to give early priority to received goods in the flow of supplied spare parts. Spare parts on remainder order and with stock level zero should be the first ones to be unpacked and controlled. This information will come from another information system, a stock control system (STO.CO). Spare parts on remainder order should never be placed in the store. They should be picked from the "supply line".

An Action Diagram describes reception and unpack of spare parts (figure 2). Spare parts can come from own factories or external suppliers. After receiving spare parts the warehouse worker registers this reception in the SPSC system. The foremen will control the continued flow through unpack to storage. In dialogue with the SPSC system they will request an unpack basis. These unpack bases are given to warehouse workers who unpack and control the goods. This is further described in the next Action Diagram (figure 3). In the quality control defects in the spare parts can be discovered. Such spare parts are taken aside.

After quality control the warehouse workers put spare parts which are approved in a waiting line for storage. A warehouse worker register the result of the unpack and control activities. This has been temporarily noted at the unpack basis during the different controls.

The foreman using the SPSC system will select, according to the priority list on the screen, spare parts to be delivered (if remainder order). The other spare parts will be put in stock also according to priorities. The warehouse worker will after picking or storing register these actions in the SPSC system.

Some abbreviations used in the diagrams:

- Stock Control system = STO.CO
- Spare Parts Supply Control System = SPSC
- Warehouse Worker = W Worker
(Own factories) \rightarrow \textit{Supplied Spare Parts} \rightarrow \textit{SP receiving (W Worker)}

(External suppliers) \rightarrow \textit{SP receiving (W Worker)}

(Foreman) \rightarrow \textit{Priority request} \rightarrow \textit{Priority check (SPSC)} \rightarrow \textit{List of priorities} \rightarrow \textit{Request (Foreman \rightarrow SPSC)}

(STO.CO) \rightarrow \textit{Remainder orders} \rightarrow \text{Stock level = 0} \rightarrow \text{POSS} \rightarrow \text{POSS}

(St0.CO) \rightarrow \textit{Delivery note} \rightarrow \textit{Identification, register receiving (W Worker \rightarrow SPSC)}

Unpack basis \rightarrow \textit{Unpack (W Worker)} \rightarrow \textit{Quantity control (W Worker)}

Supplemented unpack basis \rightarrow \textit{S2} \rightarrow \textit{Quantity controlled Spare Parts} \rightarrow \textit{S2}

\textit{Figure 2 Example of an Action Diagram (S1)}
Figure 3 Example of an Action Diagram (S2)
4.3 Some methodological comments to the example

The Action Diagrams describe different activities; some actions involve material handling and others are only administrative ("information handling"). The different actions are described by an action expression (WHAT) and also, afterwards in parentheses, by the doer (WHO). The doer can be organizational actors like foremen and warehouse workers. But a doer can also be an information system, like SPSC. In some instances action is performed in a combined way by several doers, e.g. a dialogue with the foreman and the SPSC system.

The SPSC systems performs several functions in this supply activity. These different actions are identified and separately described. They are also descriptively contextualized. Each IS function is described together with those other actions to which it has specific relations with. Each IS function is described in its right action context; in the context which it belongs to.

The Action logic in the supply activity is also expressed in the Action Diagrams. The sequence of different actions is possible to follow in the flow oriented diagrams. Alternative actions are described (OR) and conjunctive actions (explicitly by AND, or mostly implicitly by the existence of several action objects). Certain conditions can be expressed (IF remainder order). Some actions are only occurring sometimes (an unspecified condition). These contingent actions/action objects are expressed through POSS (an abbreviation of possibly occurring). Trigger of actions is expressed with an arrow. No example of interruption of actions can be found in these diagrams (figures 2 and 3). Interruption is expressed in a similar way as triggering. The arrow is substituted by the word STOP. Parallel actions are actions occurring independent of each other and possibly concurrent. Such actions can be identified when reading the flow and relations in the diagrams. If there is a need for synchronization this has to be expressed in a particular way.

The two diagrams are related to each other. Simple connectors show how they are related.

4.4 Action Diagrams in methods context

Action Diagrams are an important part in Contextual activity modelling, i.e. a method for activity development and information requirement analysis. This method is called CONTACT. In Sweden the method has been called SIM, which originally was an abbreviation for Speech act based Information Modelling [8].

Action Diagrams can be used in feasibility study/change analysis which should precede a decision on development of information systems [13, 14]. Action Diagrams can be used both for modelling existing and future activities.
The CONTACT method for activity and information requirement analysis consists of different modelling techniques. Action Diagrams are an important part of the CONTACT method, but there are other parts in this method too. Before using Action Diagrams it is important to have a clear statement of activity goals for the activity and information system. An overview of the activities should be depicted; e.g. an overview Activity graph (ISAC).

Through Action Diagrams different IS functions are identified and delimited. These IS function must be further modelled. Each such IS function is described (here using hierarchical decomposition) in a separate System Function Diagram. The notation for System Function Diagrams are nearly the same as for Action Diagrams. Different sub functions in these diagrams are later described. The selection and inference rules of these sub functions can be described in a semi-formal way (using "activity language") or a formal way. Parallel with this work the information contents and structure must be described (information modelling and concept analysis). The pragmatics from Action Diagrams should also be more specified; the communicative character of different IS actions and their intended effects in the activities should be stated more exactly. Here we take a different approach to SAMPO which is another modelling method within the language action tradition [1]. In SAMPO the different communicative functions - the so called illocutionary logic [24] - are explicitly modelled in graphical models. Confer also earlier attempts to improve ISAC graphs in this direction [9, 12]. In the CONTACT method we model this issues in separate descriptions. The Action Diagrams should have rich semantics but not semantics in overflow.

Action Diagrams form a basis for a more detailed specification of the IS. The contextual character of Action Diagrams will have decisive consequences for the detailed requirement analysis process.

5. SOME PRACTICAL EXPERIENCES USING THE METHOD

The method was developed and first used in 1986. It has been used in many different projects and applications. It has been used both in action research [17] and "normal" projects. Experiences gained from various projects have been used for improvement of the method.
I summarize some experiences

- possibility to unravel and express complex activity circumstances
- quick identification of unclear functionality in future activities
- very precise discussions concerning the activities and the role of IS
- users have (without any education in the method) been able to actively participate. The level for interpretation competence is low due to the intuitive intelligibility of the diagrams.
- some systems analysts accustomed to top down methods seem to have some initial difficulties to relearn and apply a contextual thinking.

In our research on the method we have transformed several models, which were made according to other methods (mainly the ISAC method), to Action Diagrams and compared the results. The Action Diagrams offer a more elaborated semantics. The main advantage of this extended expressability is, according to the author's conclusions, a better illustration of the dynamics of and interaction between IS and different parts of the activities.

In IS analysis and modelling there is often a difference between what on one hand is said and thought and what on the other hand is documented in explicit models. There might be a difference between the performed analysis and the recorded descriptions. Through the deeper semantics of Action Diagrams there is a greater possibility to really express the performed analysis in this graphical models. If there are discussions and design considerations about sequences, alternatives, triggers and conditions for different actions, these can be documented in the models. Our comparative analysis shows that the ISAC notation leaves a greater difference between performed and recorded analysis.

6. COMPUTER-BASED TOOLS FOR METHOD SUPPORT

Data flow diagraming is one of the most widespread notations in the area of information requirement analysis; at least if one looks at the most CASE tools currently sold. Many CASE tools contain data flow diagrams as notation. This has perhaps led to a certain standardization of systems development. Of course there is a need for standardization in systems development. The knowledge of this complex area is however evolving quickly so there is a also great need for constant innovation. Coping with this need for innovation and new methods for systems development (like the CONTACT method) implies easily adaptable CASE tools. There is an emerging interest in the concept of a CASE shell [2, 4, 25]. Using a CASE shell environment you can more quickly develop a CASE tool adapted to a specific method than using a conventional programming language.

How about computer-based tools for Contextual activity modelling and Action Diagrams? We have chosen a CASE shell strategy. There is a prototype version of an Action Diagram tool developed at Jyväskylä University with their CASE shell MetaEdit [25]. At the moment
we are, at Linköping University, developing two Action Diagram tools. We are using the widespread commercial product Excelerator/Customizer. We are also using Ramatic, a CASE shell from the Swedish research institute SISU [2]. We are planning to report from this comparative study.

One rationale for using a CASE shell strategy is that we want to investigate the potential of this software technology [4]. How easy is it develop your own CASE tool? How powerful will such tools be?

7. CONCLUSIONS

It is important that information systems are well integrated and supportive to their activity environment. Therefore it is important to model the dynamics and interaction of IS and activities. This is the main rationale behind this approach of Contextual activity modelling. The value of IS lies in the support it gives to surrounding activities. It has no value of its own.

Describing IS in relation to the activities can be done in several ways. The author has distinguished two main descriptive approaches:
- describing IS as a unified whole
- describing IS as several contextualized actions

Seeing IS as a unified whole means that you emphasize it as a unified doer. In one respect it is a unified doer of course. An information system will consist of integrated software. But is this the best way to view it when performing activity and information requirements analysis? The arguments here is that it is important to visualize the action character of IS. When designing an activity pattern (including an IS) one must be able to model the dynamics of this activity pattern. To capture this dynamic character one must apply an appropriate approach for analysis and description. When taking a unified whole approach, as in SA and ISAC, the doer aspect comes in foreground and the functional/action aspect comes in background. Contextual activity modelling is taking the opposite approach: Putting the action aspect in foreground and the doer aspect in background.

Stressing the action aspect, and by that explicitly modelling it, has two important consequences. The action logic of the activity pattern can be more thoroughly explicated and recorded. The effects of IS action (its communication) can be more deeply analyzable and showable. When taking such design issues into explicit consideration it should improve the chances to produce IS that are well integrated in and supportive to their environmental activities.
One important difference between these two modelling approaches is that
- the unified whole approach stresses the *separation* between IS and activities and
- the contextualized action approach stresses the *integration* of IS and activities

The author's conclusions is that it is preferable to view the action character of the IS; to consider it performing several different actions; and describing each action type in its action context. When taking this approach there are greater possibilities to describe the dynamic character of this IS - environment interaction.

In this concluding section I will also recall the arguments for hierarchical decomposition mentioned in section 2 above. One argument is that different sub descriptions should be related to each other in a controllable fashion. The contextual descriptions of Action Diagrams can be related to each other in a controllable way. The other argument was the need for both overview and detail. Action diagrams fulfill the need for detail. For overview other more appropriate graphs should be used.

I have used the well-known methods Structured Analysis/Data flow diagrams and ISAC/Activity graphs as reference for my discussion. I will end this paper with a summarizing table (figure 4) over similarities and differences between the three methods.

<table>
<thead>
<tr>
<th>Form of description</th>
<th>Data flow diagrams (SA)</th>
<th>Activity graphs (ISAC)</th>
<th>Action diagrams (CONTACT)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analytic/descriptive approach</td>
<td>graphical</td>
<td>graphical</td>
<td>graphical</td>
</tr>
<tr>
<td>Scope of activity description</td>
<td>hierarchical decomposition</td>
<td>hierarchical decomposition</td>
<td>contextual (hierarchical permitted)</td>
</tr>
<tr>
<td>Description of IS</td>
<td>information flow</td>
<td>information &amp; material flow</td>
<td>information &amp; material flow action logic</td>
</tr>
<tr>
<td></td>
<td>unified whole</td>
<td>unified whole</td>
<td>several contextualized actions</td>
</tr>
</tbody>
</table>

*figure 4 Comparison between activity oriented IS modelling techniques*
ACKNOWLEDGEMENTS

During research on Contextual activity modelling the author has gained from cooperation with several people. I will especially mention Annie Röstlinger. The author is grateful to James Nyce, Ian Herbert and Toomas Timpka for good comments on this paper.

Parts of this research has been performed with financial support from The Swedish National Board for Industrial and Technical Development (TTYP Program) and The Swedish Council for Research in the Humanities and Social Sciences.

REFERENCES


development - A systematic approach, Prentice-Hall, Englewood Cliffs

Management, 12, 1987

dynamics in information systems, in [26]


A flexible graphical environment for methodology modelling, Dept of
Computer Science, University of Jyväskylä

systems, North-Holland, Amsterdam

approach to the dynamic modelling of information systems, in [26]

new foundation for design, Ablex, Norwood

[29] Wittgenstein L (1953) Philosophical investigations, Basil Blackwell,
London

APPENDICE

NOTATIONAL RULES FOR ACTION DIAGRAMS

- Information (oral, written or electronic)
- Material object(s)
- Communication (information flow)
- Material flow
- Action with doer
- Sequence of actions
- Trigger by communication
- Trigger by time

Action (Doer)
APPENDICE

NOTATIONAL RULES FOR ACTION DIAGRAMS

Interruption of actions

OR

Alternative actions

OR

Conjunctive actions. This notation is mainly used together with alternatives.

AND

Conjunctive actions are usually described implicitly by the occurrence of several action objects.

AND

IF/WHEN condition

Condition for action/action object

POSS

Contingent action/action object (possibly occurring)

Action objects may be downplayed, only if the meaning of them are defined by the context and it enhances overview.

Connectors; relate action diagrams to each other

These are the most important rules for action diagrams. There are other rules for special situations.