Information Systems Actability Engineering

— Integrating Analysis of Business Processes and Usability Requirements

Pär J. Ågerfalk[1, 3], Göran Goldkuhl[2, 3], Stefan Cronholm[4]

Research Group VITS

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

[2]Dept. of Informatics, Jönköping International Business School
P.O. 1026, SE-551 11 Jönköping, Sweden
ggo@ida.liu.se

[3]Centre for studies on humans, technology and organization (CMTO)
Linköping University, SE-581 83 Linköping, Sweden

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

Abstract

When developing computerised information systems it is common to begin with some kind of business modelling to get an understanding of the planned systems environment. Different usability factors are also common to consider when the user interface is being designed. What is not so common is to take advantage of each of these activities when the other one is performed. By applying a language action perspective on the design of business processes and information systems this paper shows how to integrate elicitation, analysis and validation of business requirements and usability requirements. By viewing human-computer interaction as a crucial part of doing business, the integrated analysis of requirements concerning business processes and usability becomes analysis of information systems actability requirements; i.e. Actability Engineering.

1 Introduction

When developing information systems it is common to begin with some kind of business modelling in order to get an understanding of the business context in which new systems are to be used (e.g. Jacobson *et al.*, 1995; Bubenko & Kirikova, 1999). What is not so common is to make explicit use of the results of such modelling when defining system requirements, especially not requirements concerning human-computer interaction and usability. Information systems are used to perform and support business actions and system interaction models therefore ought to be derived explicitly from business models.

Many approaches to information systems engineering do take the connection between business and information systems into account (e.g. Jacobson *et al.*, 1995; Jackson & Twaddle, 1997). At least in the sense that they make use of conceptual business modelling when defining requirements for systems' information content as well as derive interaction points from identified business tasks. What they seem to miss is that the intentional actions forming business tasks is crucial when designing user interfaces. An information system (IS) that is unclear about what business actions it is able to perform, in interaction or automatically, is most likely unusable.

On the other hand, available models for usability design (e.g. Lövgren, 1993; Schneiderman, 1998) seem to be too narrow and focus too much on the interaction *per se*, without appropriately and sufficiently relating it to the business context. Usability is often thought of as a property of the artefact in terms of, for example, relevance, learnability, flexibility, and efficiency (Löwgren, 1993; Preece *et al.*, 1994). Relevance is defined by Löwgren (1993) as: "*… how well the system serves the users' needs*". We believe that the users' needs must be judged in the light of the social context in which the users act. This context awareness is emphasised within the field of computer supported co-operative work (CSCW), partly as a reaction against the system focused usability of usability engineering (Löwgren, 1995). There is however more than collaboration among actors and co-ordination of activities that form the social context of a business. In order to design for usability we must have a proper understanding of the business: its action structure as well as internal and external actors and agents and their professional language use.

We thus have identified a requirements engineering (RE) gap, or at least mismatch, between business modelling and systems modelling (see fig 1). In order to bridge the RE gap we propose that 1) human-computer interaction should be regarded as a crucial part of doing business and its design should therefore be based on business modelling and 2) business processes should be designed with usability in mind. In other words, we would like to extend 1) “business modelling” to incorporate human-computer interaction aspects and 2) “usability” to incorporate design of business processes.

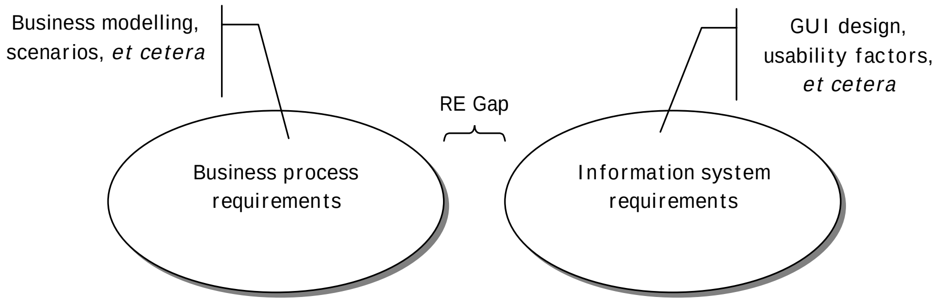

Figure 1: The requirements engineering gap

We will in this paper discuss how a reconciliation of business modelling and usability design can be performed to create what we call actability – the most important quality of any information system. An information system’s actability is thought of as *its ability to perform actions and to permit, promote and facilitate users to perform their actions both through the system and based on messages from the system, in some business context*. We propose an integrated approach with the notion of “communicative action” as the integrator. By applying a language action perspective, the design of business processes becomes design of communicative action structures. Since business actors and agents perform many communicative actions by use of information systems, information systems requirements become crucial parts of business requirements. We refer to this integration as *Actability Engineering*.

2 Communicative Action as Integrating Concept

Information systems can be viewed in different ways. Many approaches and theories within the Information systems area are based on a strict representational view of information. Information systems are considered as “containers of facts”. We do not deny that information systems contain information as descriptions of reality, but this is a too restricted view of the character of information systems. Instead we argue that information systems are vehicles for communication and action among people and organisations (Goldkuhl & Ågerfalk, 1998). This is following the language action perspective on information systems (e.g. Goldkuhl & Lyytinen, 1982; Winograd & Flores, 1986), which is based on speech act theory (e.g. Austin, 1962; Searle, 1969). The key message of speech act theory is that an utterance consists of both a “propositional content” (describing the world) and an ”illocutionary force” (i.e. an action mode). The illocution used expresses the action performed through speech and thus the type of relationship established between speaker and hearer.

Computerised information systems are parts of organisations. Commercial enterprises make business with actors (customers and suppliers) in their environment. Making business means performing different business actions. There are different business actions performed by a supplier as, for example, exposure of products, offering, con-firming orders, delivering products, and invoicing. In Business Action Theory (BAT) such different actions are described and related to the actions of customers (Goldkuhl, 1996; 1998). BAT is based on speech act theory and business relationships theories (e.g. Axelsson & Easton, 1992) and is used as primary theoretical basis for our view on information systems and approach to requirements engineering. Another such theory, popular, but also criticised, within the field of CSCW, is the Action Workflow approach (e.g. Denning & Medina-Mora, 1995). One aim of BAT is to overcome much of the criticisms directed towards Action Workflow, such that it provides a to simple picture of doing business and that it leads to increased control of work tasks (cf. Goldkuhl, 1996; Ljungberg & Holm, 1996).

Information systems are used to perform different business actions in an organisation. An IS can 1) be used as a support for human actors to perform action (”the support view”). An actor ”reads” the IS and becomes informed about earlier actions and other important action prerequisites. To be able to give such a support an IS must contain a memory of earlier actions and other necessary conditions. But an IS is not restricted to this. It can also 2) perform predefined actions automatically (as, for example, sending out invoices to customers). It can also 3) be used to perform actions interactively. Many actions seem to be performed by a user and IS in interaction. For example an order is created when a sales person, based on expressed desires from a customer, fill out an order form in an interactive IS. The order creation, as an act, is established interactively through the IS.

This action view can be elaborated into a definition of "information system" (cf. Goldkuhl & Ågerfalk, 1998): *An information system consists of an action repertoire (action potential) and an organisational memory intended for action. An information system can perform actions automatically or actions can be performed by user and system interactively through the system*.

This action theoretical standpoint emphasises that organisations perform actions when making business. Information systems are vehicles in such organisational action. Not only by giving informational support for action, but also by performing actions interactively or automatically.

Basic features of an IS must thus be its ability to perform actions and to permit, promote and facilitate users to perform their actions both through the system and based on messages from the system, in some business context. We call this the actability of an IS. We can distinguish between direct business actions (which are directed towards customers) and indirect business actions (which are made as a prerequisite for direct business actions). IS actability may concern both direct and indirect business action. IS actability is thus strongly related to the ability of the company to establish mutually good agreements with customers and to satisfy customers' needs and demands of deliveries of appropriate products.

3 Interactive Usage Situations in Business Processes

Nowadays, there is great interest in process orientation when developing organisations and information systems (e.g. Davenport, 1993; Jacobson *et al.*, 1995). A process view on organisations implies a horizontal view on how the organisation creates products (for example goods and services) with benefits for customers. A business process consists of activities ordered in a structured way with the purpose to produce valuable results for customers. Different persons within the organisation (and even outside) can perform such activities. Customers must perform certain actions as, for example, inquiring and ordering. Some actions of the organisation can be performed by computerised information systems as been described in section 2.

Modelling and designing business processes means describing different actions that should be performed within a business process. The process logic should capture how different actions are related to each other and different "firing conditions" for actions.

There exist several approaches to describe business processes such as, for example, scenario diagrams, activity diagrams and data flow diagrams. To describe business processes we propose Action Diagrams of the SIMM family of methods (Goldkuhl, 1992; 1996; Ågerfalk & Goldkuhl, 1998) to be used, due to insufficient semantics in other formalisms.

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

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

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

Figure 2 and Figure 3 show example Action Diagrams describing two connected business contexts. The first diagram (PurInq, see Figure 2) describes how feasibility checking regarding raw material is performed at a company referred to as "The Paper Mill". At The Paper Mill feasibility checking is done to ensure that enough raw material is either in stock or ordered before an enquiry or order from a customer is acknowledged. The second diagram (PurAck, see Figure 3) describes how purchase order acknowledgements from suppliers and deliveries are received and registered. In addition, the second diagram also shows an activity "Move raw material" that might occur during raw material receiving, but which might as well occur in isolation. This exemplifies the freedom to choose arbitrary boundaries for business contexts during analysis.

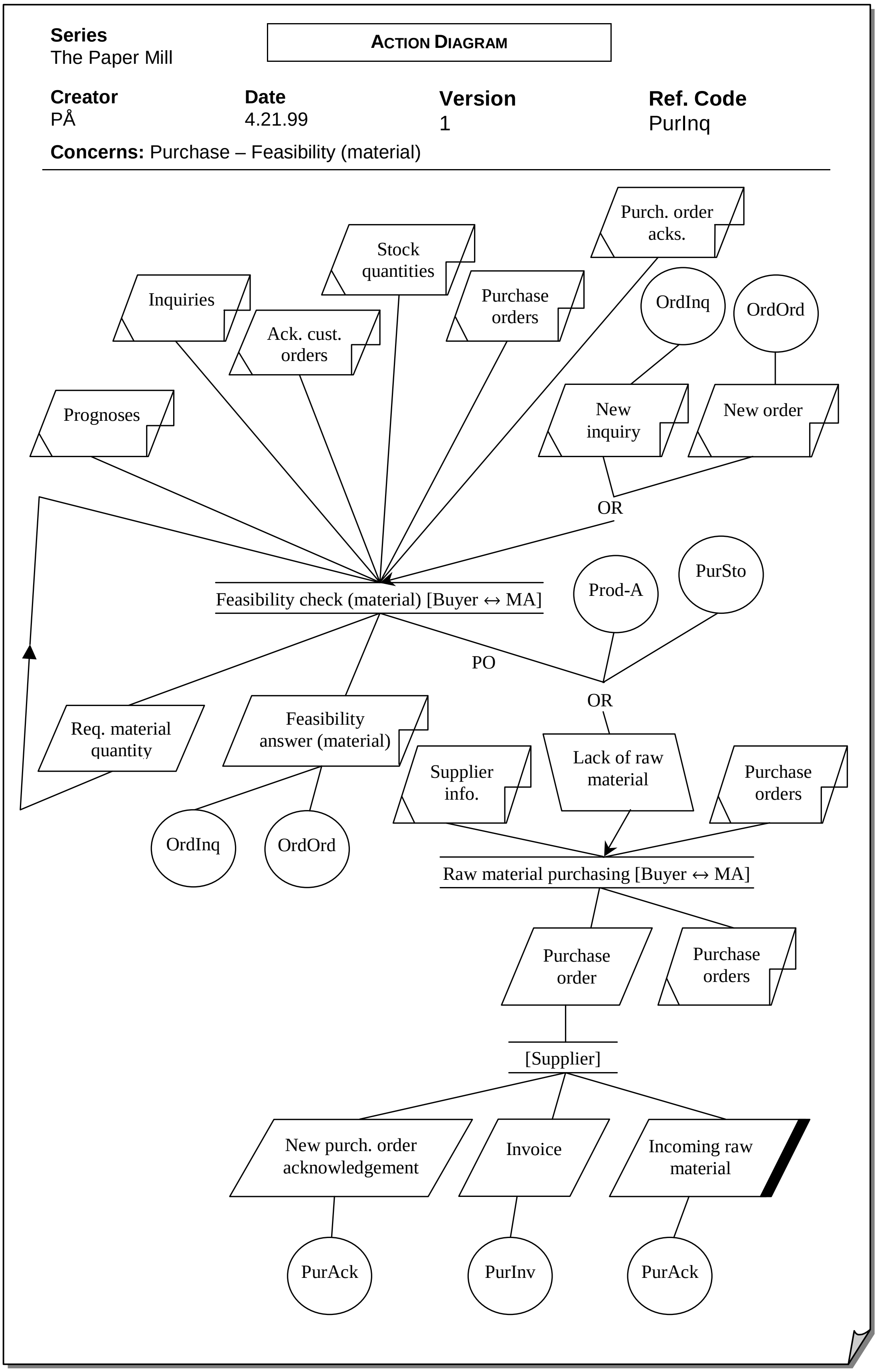

Figure 2: Example Action Diagram I

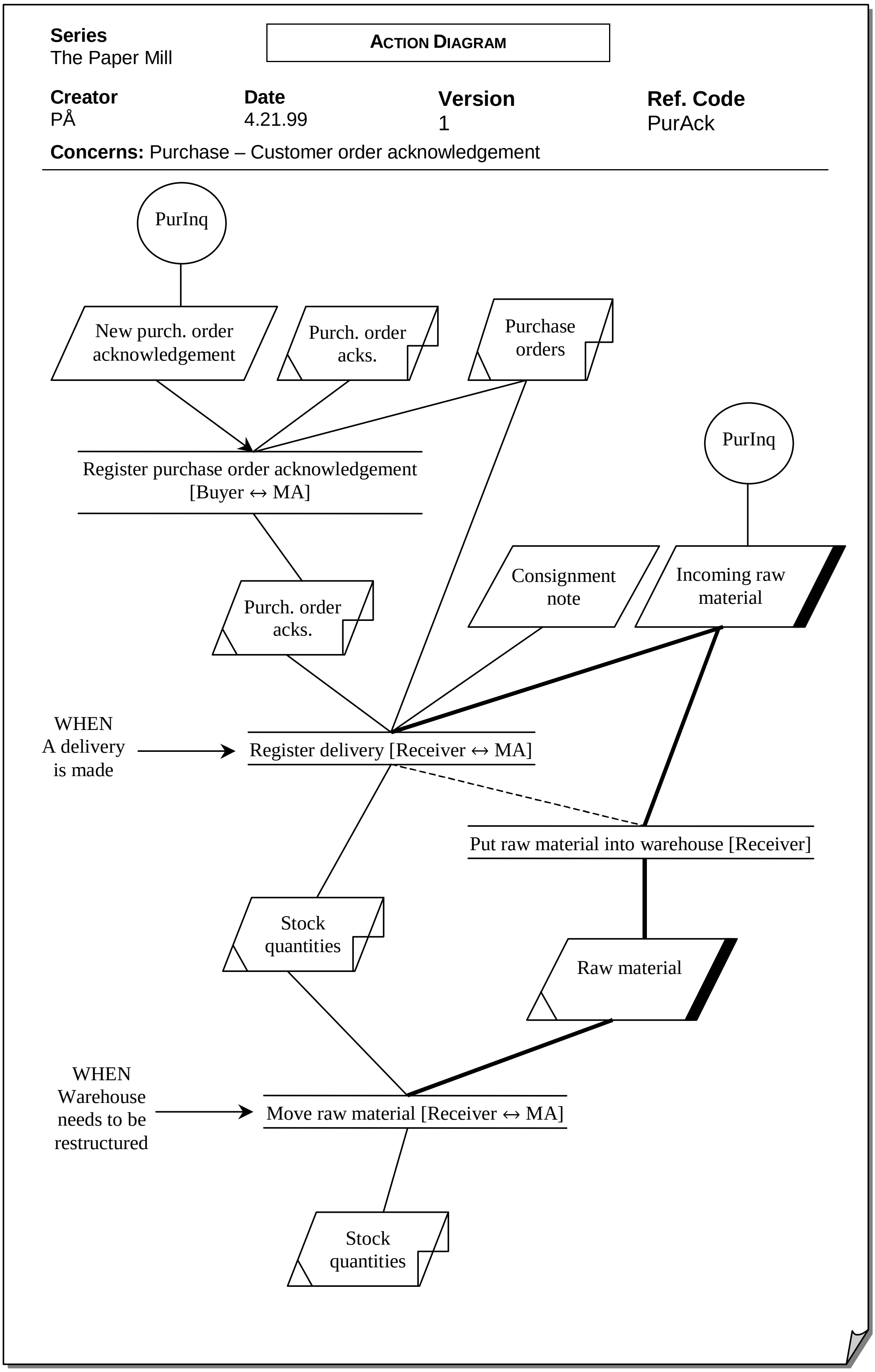

Figure 3: Example Action Diagram II

The Action Diagrams give a possibility to model and design the business and its information systems as an integrated whole. The actions of different performers — human actors and/or information systems (internal and/or external to the organisation) — are described as a whole. Hence, we use Action Diagrams to identify and delimit IS actions, which are described as integral parts of the business process. To say that IS actions are derived from the business process design is one way to put it. That business process design includes design of IS actions is another equally correct. This counts for both interactive IS action (performed together with a user) and automatic IS action (performed by the computerised IS itself). The Action Diagrams also show actions performed as a consequence of IS action. These IS-related actions form what we call usage situations, which can be classified as 1) interactive 2) automatic or 3) consequential (see Figure 4).

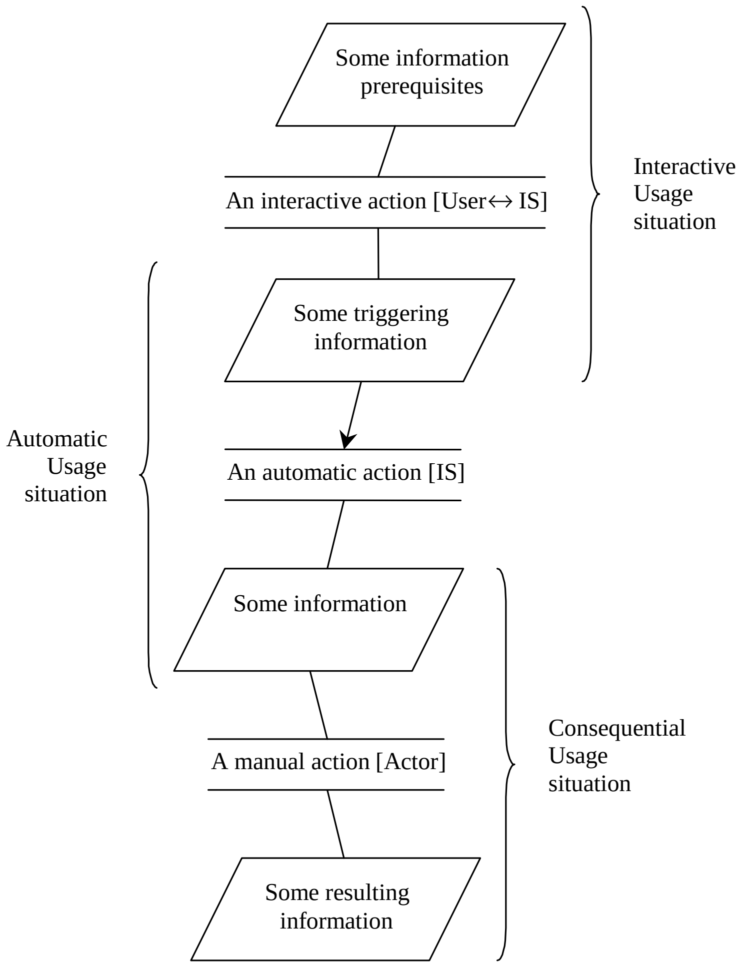

Figure 4: Interactive, automatic and consequential usage situations

Business process modelling with Action Diagrams is a way to ensure that the IS becomes an integrated part of the business process and not an "isolated thing" outside the process. This kind of business process modelling is also a way to identify and delineate appropriate interactive usage situations (see section 4.2.3). We refer to these as "Interactive usage Situation Proposals" (ISPs) which form a basis for analysis of actability requirements. How this is done is described in sections 4 and 5.

4 Understanding Actability Requirements

In this section we present a framework for actability requirements engineering; i.e. for how human-computer interaction modelling and evaluation could be performed in order to achieve high actability. The main purpose of such engineering is to elicit and analyse actability requirements based on the action structure of the business process being designed. Another important purpose is to enable traceability of requirements from business model to detailed system models.

4.1 Performing Speech Acts…

To say something is, according to Austin (1962), to perform three simultaneous acts: one locutionary act, one illocutionary act, and one perlocutionary act. The locutionary act is the act of uttering certain words in a certain sequence with a certain sense and reference (i.e. meaning). When performing a locutionary act an illocutionary act is also performed simultaneously, i.e. we do something by speaking, for example, making a promise, warn, *et cetera* (cf. section 2). Doing something will probably also (usually) have some effect on a hearer of the utterance, e.g. changing her knowledge or causing him to act in response. Causing these effects is to perform a perlocutionary act.

Searle (1969) adopted Austin's concepts of illocutionary and perlocutionary acts but refined the concept of locutionary act, which lead to a notion of speech acts as consisting of four different acts:

a) Uttering words = performing *utterance acts*.
b) Referring and predicating = performing *propositional acts*.
c) Stating, questioning, commanding, promising, *et cetera* = performing *illocutionary acts*.

d) Causing effect in hearers = performing *perlocutionary acts*.

Note that the first three parts (a, b and c) are not separate things that a speaker does simultaneously. Likewise, a and b are not means to achieve c. Rather, as Searle (1969) puts it: "*utterance acts stand to propositional acts and illocutionary acts in the way in which, e.g., making an "X" on a ballot paper stands to voting*".

However, following Norman (1988), we argue that before being able to perform an utterance act the speaker has to form the goal and intention of the utterance as well as formulate the sequence of words to utter (specifying the action in Norman's terminology). This is to say that based on the desired perlocution (effect) the speaker chooses illocution (what to do) and formulates the utterance (word sequence) concerning something (propositional content) in her mind before the actual uttering; i.e. thinking before talking.

We believe that an IS should support both the formulation and the uttering (execution) of speech acts in order to be actable.

4.2 … through information systems

Before going into details of interaction analysis we will first establish some terminological conventions that we believe are more comfortable to use in the context of information systems than those of "pure" speech act theory.

By action elementary message (ae-message) we refer to the result of performing a speech act. Thus, an ae-message is what is being sent through an IS when performing an elementary action (e-action). An ae-message consists of a propositional content and an action mode corresponding to the propositional act and the illocutionary act. The propositional content of an ae-message consists of (at the type level) one or more entities and attributes with possibly relationships among them (referring and predicating). Since there is a one-to-one correspondence between ae-messages and e-actions, the action mode of an ae-message implies what communication function the e-action serves; i.e. the kind of relation created between the communicator (the speaker) and the interpreter (the hearer). The process of creating an ae-message is referred to as formulate the ae-message and the act of uttering it as execute the e-action or, alternatively, send the ae-message. Some agent (human or artefact) acting on the communicator's behalf might perform the e-action. The perlocution is referred to as communication effect.

Basically, Actability Engineering is concerned with three questions, see Figure 5.

- Which AE-messages have to be sent to achieve the desired business effects, i.e. what E-actions have to be performed to produce the desired output?
- For each AE-message, what is it's: propositional content; communication functions; communication effects; communicator and agent; intended interpreters; and, what information from previously sent AE-messages is required to formulate this one?
- What computer support is required to formulate and send the AE-messages?

Figure 5: The three questions of Actability Engineering.

As indicated in section 3, we suggest that the interaction analysis is performed on a "per usage situation (ISP) basis" as an integral part of business modelling. This is necessary in order to cope with complexity (by analysing the parts of the system separately) and still maintain a holistic view of the business context, in which the interaction is taking place, i.e. part of bridging the RE gap. Furthermore, we suggest that the analysis of each usage situation is performed on a "per ae-message basis", for similar reasons.

Thus, an interactive usage situation consists of formulation and sending of one or more ae-messages. The process of formulating an ae-message should be supported by the IS and can be viewed as a dialogue between the user and the IS. Such a dialogue consists of interaction cycles of user actions, IS actions and user interpretation acts, referred to as elementary interactions (e-interactions), in accordance to the Elementary InterAction Loop (EIAL), further discussed subsequently (section 4.2.2). Before any user actions is performed, the IS might perform some initial IS actions that give the user initial information about possible kinds of e-actions to perform and/or suggested content of an ae-message to send. Both user actions and IS actions result in a special kind of messages referred to as interactive messages (ia-messages) used "ISP-internally" during formulation of an ae-message.

A special kind of e-interaction is the one triggered by the user action that sends the actual ae-message and thus executes the elementary action. Figure 6 illustrates these concepts with ia-messages being sent between the user and the computer, and the ae-message, sent through the IS, causing some business effect(s).

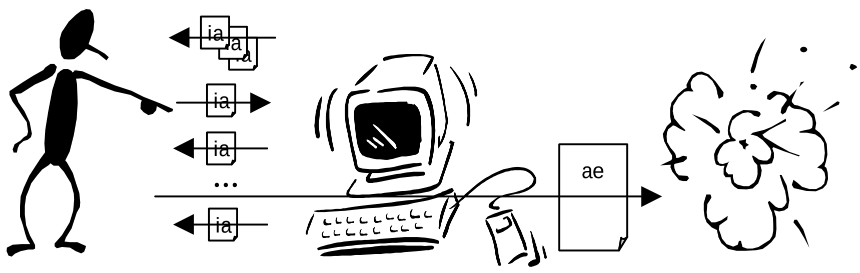

Figure 6: Action elementary messages is formulated during e-interactions in dialogue with, and sent trough, the IS in order to perform some business effect(s)

Messages can be visualised as documents. An "Invoice" document used to command customers to pay, for example, would typically be used to carry the ae-message "Customer invoice" and thus corresponds to the e-action "Send invoice to customer".

We can distinguish three categories of document types that call for somewhat different treatment. There are paper documents, such as reports and lists, there are electronically transmitted documents, such as EDI-documents and there are interactive (screen) documents, which permit direct manipulation on the computer screen, for example windows and forms. These document categories obviously relate to the three kinds of usage situations discussed in section 3. Since usage situations overlap (see Figure 4), all three categories must be considered during interaction analysis, though.

Let us illustrate by an example. To create the effect of making a customer to pay (effect), a business firm (communicator) might send an EDI-document (electronic document interchange) "Customer invoice" to command (action mode) the customer to do so (function). Some office clerk (agent) performs the e-action of sending the invoice during an interactive usage situation. To perform the e-action the interactive document "Invoice" is used. "Invoice" consists, among other things, of a button labelled "Send invoice" used to execute the e-action. At the start of the interactive usage situation, the IS retrieves relevant information (propositional content) about the customer and the customer order that the invoice concerns and presents it in the interactive document (initial IS actions). But, before executing the e-action, the invoice has to be manually checked for correctness and, perhaps, some discounts need to be manually chosen given some different options (e-interactions). Finally, when the "Send invoice" button is clicked, the EDI-document is sent to the customer and a pop-up window notifies the user about this (i.e. the special kind of e-interaction that executes the e-action).

4.2.1 Interactive Screen Documents Action Potential and States

All documents have an action potential. In the case of interactive screen documents the action potential is materialised as the functionality offered to the interacting actor (user) at any given point in time (i.e. the possible e-interactions to initiate) by, for example screen items to click. An interactive screen document's action potential varies over time as the result of 1) the type of ISP at hand and 2) the state of that ISP. A "Product information" interactive document might, for example, offer different action potential during ISPs "Create customer order" and "Stocktaking" respectively. During "Create customer order" the same document might also require that some customer details have been entered (a certain state is reached) before the order can be registered. Such changes in action potential can and should be visible to the user (cf. Norman's (1988) discussion of "affordance"), which could be done by, for example, disabling buttons that cannot be used in a certain state.

4.2.2 The Elementary InterAction Loop

To analyse human-computer interaction we use the concept of e-interaction. Decomposing user actions until further decomposition would introduce terminology that is not part of the business' ontology, i.e. computer terminology such as "move the mouse", delineates e-interactions (cf. Graham (1998) on "task objects"). Basically an interactive usage situation is constituted by a set of e-interactions with some sequence restrictions applied.

Each e-interaction (initiated by a user action) is matched by one ore more actions performed by the system. That is, the actor performs a user action whereupon the system answers by performing its corresponding IS actions. After the IS action(s) are performed the actor interprets what the system has achieved. This sequence of actions constitutes an Elementary InterAction Loop (EIAL; see Figure 7).

During the execution of an instance of such a loop the interactive situation is in a transient state and the loop thus has to be completed before a new e-interaction can occur. Actually there are three different states involved; an "initial state" (s_0) before the user action, a "waiting for system to respond state" (s_1) between the user action and the IS action(s) and an "IS action accomplished state" (s_2) after the IS action(s). It might be the case that the actor has to switch to another e-interaction in order to complete the interpretation act, though. For example to check some details in another document. Therefore each e-interaction might be infinitely recursively expanded. Such sub-interactions must be permitted but analysis must explore the possibility that two or more e-interactions might have been regarded as one, in which case these should be separated and treated independently.

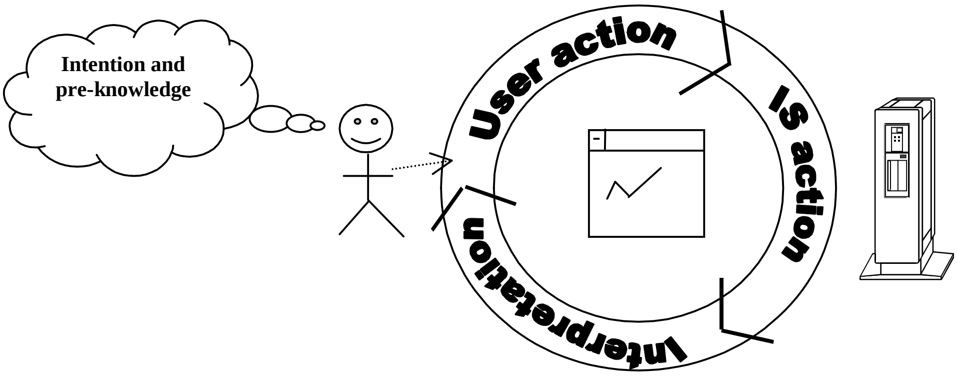

Figure 7: The Elementary InterAction Loop

Note that interactive usage situations are derived from business process modelling and how the EIAL thereby connects business requirements with functional requirements and usability requirements of the planned IS; i.e. actability requirements.

4.2.3 Modelling and Validation of Actability Requirements

As mentioned previously, interactive usage situation proposals (ISPs) are derived explicitly from business modelling. An ISP is defined as a primitive sequence of business actions that consists of all interactive actions, and intermediate manual and automatic actions, that are performed adjacent in time at the same place by the same performers (cf. Holm & Ljungberg's (1996) notion of "conversations"). ISPs are thus found by selecting arbitrary interactive IS actions in the Action Diagrams and traversing the flow of activities, in both directions (preceding and succeeding), until all adjacent activities, in each direction, not conformant to the definition is found. The ISPs are then used as a basis for a more detailed analysis concerning the interactive actions and the design of interactive documents. The "performer part" of interactive actions is thus focused. We turn our attention to the information that flow (the messages sent) between the actor and the system, its origin and form, to analyse functional requirements and requirements concerning usability in the context of business processes.

Interaction analysis is preferably performed with a mixed analytical and experimental approach in a dialectic manner. Analytical models are continuously validated against document prototypes and prototypes are continuously verified against analytical models (Goldkuhl & Ågerfalk, 1998). This way both consistence and relevance is achieved in a way hardly obtained by using only one of the approaches exclusively (cf. Mathiassen *et al.*, 1995).

Both modelling and validation of requirements are performed based on the "three questions of requirements engineering for actability" as discussed previously, see Figure 5. The ae-messages to be sent in each interactive usage situation can be identified by looking at the results of the corresponding actions in the business model (the Action Diagrams). These results are the means to reach desired effects of the interactive usage situation. For each ae-message there should be a propositional content and an illocutionary force that support those effects. Furthermore, there should be possible e-interactions and initial IS actions that permit, promote and facilitate the formulation of those ae-messages. To ensure this, different usability factors, such as "easy to learn" and "easy to use" should be considered and, for example, interaction style guides could be used as an aid (cf. Preece *et al.*, 1994; Card & Anderson, 1987; Monk *et al.*, 1993; Nielsen, 1993; Lauesson & Younessi, 1998).

During the Actability Engineering process both the contents and demarcations of the ISPs are likely to change (which is why they are referred to as proposals). Results from a performed case study shows that earlier assumptions about the business action structure might have to be reconsidered and interaction analysis thus becomes a valuable source of information also to business process modelling. This is an example of the need of integrating business modelling and interaction design. The necessity to formalise the action structure reveals information hard to elicit only by interviewing and observing actors. By putting actors in front of the future system they feel the need to be explicit, and they instantly see the effects of alternative business descriptions.

In the following section we will discuss how to document actability requirements.

5 Documenting Actability Requirements

In this section we will describe how actability requirements could be described in order to bridge the RE gap and to build a solid base for construction information systems with high quality actability. The proposed documenting techniques are based on the theoretical discussions previously in the paper and on the belief that RE practice is most benefited if reuse of existing modelling formalisms, extended with necessary concepts and notations, is provided.

5.1 Message and Document Definitions

For each ae-message, we propose that a Message Definition should be created. Similarly, for each document that is used to create and/or carry ae-messages, a Document Definition should be created.

A Message Definition, in its simplest form, consists of a text document stating primary and secondary communication functions, primary and secondary communication effects, communicator, intended interpreters, and performer of the corresponding e-action). Despite this information there should also be references to the ISPs and documents in which the message is created.

A Document definition, which might also be a text document, should at least state the type of document (paper, interactive screen, *et cetera*), what ae-messages the document carries and in what ISPs it is used. Furthermore, in the case of interactive screen documents, it should state which ae-messages the document is responsible for creating (assist in formulating) and sending.

5.2 I-Tables and Interactive Document Layouts

To model actability requirements, we propose Interaction Tables (I-Tables), which build explicitly on the EIAL, to be used. An I-Table is a table with three rows and three columns. The leftmost column is used for the user actions, the middle column for the state of the current interactive document(s) and the rightmost column is used for the IS actions. One I-Table is used for each e-interaction (EIAL).

<table>
<tr><th>3</th><th>User action</th><th>Document</th><th>IS action</th></tr>
<tr><td>S_0</td><td></td><td>1. Order form
2. Customer details</td><td></td></tr>
<tr><td>S_1</td><td><u>Associate customer to order</u>
{By drag 'n drop 2→1}</td><td><u>Order form</u>
Order with associated customer</td><td rowspan="2">1. Associate customer to current order.
2. Transfer information about current customer to current order (from customer file).</td></tr>
<tr><td>S_2</td><td><u>Acknowledge customer information:</u>
1. Correct customer information.
2. Incorrect customer information. (→ 2)</td><td><u>Order form</u>
Order with associated customer information visible.</td></tr>
</table>

Figure 8: Example I-Table for an IS to support a travel agency

Figure 8 shows the I-Table for e-interaction 3 within the ISP "Create customer order" from an IS to support a travel agency. From the example in Figure 8 we see how s_0 specifies what documents are to be visible in order to perform e-interaction 3, i.e. a precondition. Cell (2,1) specifies what action the I-Table is concerned with, i.e. associating a customer to an order by the use of "drag 'n drop" from the document "Customer details" to the document "Order form". Cell (2,2) specifies the state of the e-interaction after the user action. The customer is now associated to the order (from the user's point of view). Column 3 shows the actions that are to be performed by the IS in response to the user's action (in this case two consecutive IS actions). Cell (3,2) shows the response from the IS, i.e. the order form with visible customer information. Finally, cell (3,1) shows the interpretation act to be performed by the user. In this case the interpretation should yield either that the customer information is correct or that the information needs to bee updated, which leads to the performance of e-interaction 2, i.e. updating customer information as described in another I-Table (No 2).

In most cases the number of IS actions that correspond to a user action is one or perhaps two in strict sequence. Sometimes it might however be several IS actions involved and thus a need to describe how these relate to each other. In such cases a regular expression, specifying the sequence, can be augmented to column 3.

The granularity of which the documents are represented in I-Tables depends on how far the analysis has proceeded. At first, the documents are referred to by textual references (as in Figure 8). When analysis proceeds the layouts of the documents are getting more and more explicit. This evolution can (and should) be shown in the I-Tables. This is done as in Figure 9, which shows the e-interaction number 2 from the same IS as in Figure 8.

2	User action	Document	IS action
S_1	**Correct invalid customer information**	**Change customer information** Old information New information [Register change]	Update customer information.
S_2	**State fact:** Correct customer information is registered.	**Change customer information (acknowledged)**  	{RT ≤ 1s}

Figure 9: I-Table with detailed document layout

It is sometimes unnecessary to show the initial state s_0 within an I-Table, and hence the first row can be omitted, as done in Figure 9.

The language used within I-Tables is natural language such as English or Swedish. We do however use some conventions that semi-formalise the language use as described by Ågerfalk & Goldkuhl (1998). For example, usability requirements can be formulated as done in Figure 9 where "{RT < 1s}" states a requirement regarding response time.

5.3 Statecharts and Action Potential of Interactive Screen Documents

Each I-Table describes one e-interaction so there is a need to describe how these are related to each other. As described previously, the action potential of any given interactive screen document might vary over time. Since action sequence restrictions constitute the state of the document, and thereby restrict action potential, we use the UML version (Booch *et al.*, 1999) of the Statechart formalism (Harel, 1987) to model them. A related issue is the modelling of navigation paths between documents, which could be done with I-Tables but such an attempt would not be tractable due to the many possible navigation paths. Instead, this is also done with statecharts where each high-level state represents a document and each navigational action is represented by a transition. The resulting state model can be viewed as consisting of documents as high-level states where the action sequence restrictions of each document correspond to sub-states of those.

This approach to model action sequence restrictions is an extended version of Horrocks' (1999) Statechart approach to user interface construction. With Horrocks' approach there is a clear path from interface design to implementation in, preferably, event driven rapid development tools (e.g. MS Visual Basic, Borland Delphi). However, his approach lacks a clear connection to business actions, which is provided by the use of e-interactions as a source for transitions rather than Horrocks' vaguely defined notion of *event*. Note, however, that Horrocks (*ibid.*) explicitly states that his approach is not a complete user interface design approach but an approach to specifying the behaviour of the user interface.

5.4 Conceptual Modelling and Propositional Content

In order to design interactions and interactive documents there is a need to also consider the propositional content of messages. This is preferably done during a conceptual analysis based on identified ae-messages. Conceptual modelling and interaction analysis is thus performed as an integrated process (remember Searle's voting). During interaction analysis the professional language used by actors is captured and used in interactive documents. This language is then analysed by "traditional" conceptual modelling in terms of entities, properties and relationships; cf. E/R modelling (Chen, 1976) and object-oriented analysis (e.g. Booch, 1994; Blaha & Premerlani, 1998). We propose to use UML Class Diagrams (Booch *et al.*, 1999) for this purpose, which have proven useful during a performed case study.

Interaction analysis concerns interactive usage situations and hence interactive documents have been in focus of this paper. The documents used in consequential and automatic usage situations are often based on ae-messages from interactive usage situations and they are therefore normally treaded during interaction analysis. However, there is a need to "walk trough" the business model and make sure that all documents have been taken into account. There might also be a need to consider all ae-messages, documents and propositional contents as a whole and consolidate the, to some extent, fragmented document model and conceptual model of interaction analysis to system global level. Note that by use of a method tool (CASE), such as Rational Rose, it is possible to do conceptual analysis based on "parts" of an object model, for example messages, and still maintain global consistency. This is important to verify that the same concept (class) is not used with, for example, different definitions in different contexts.

6 Conclusions and Further Actability Discussion

We have argued that design of information systems should be performed with a proper understanding of the business, gained through an integrated development of business processes and their supporting information systems. When designing interaction, usability factors such as "easy to learn", "easy to use" and "interaction styles" are important to consider. We do however believe that the common notion of "usability" is to narrow since it is often perceived as dealing only with design of user interfaces. When designing communication trough an IS, the question of how to interact is, of course, important. Equally important, though, is what to communicate and why. Moreover, all three aspects must be considered in the business context where the communication is taking place. That is, a social context that is never static and fully predictable.

We use the concept "actabiltity", which is based on theories from the language/action perspective and usability, to talk about information systems use in business processes. We interpret actability as being an information systems ability to perform actions and to permit, promote and facilitate users to perform their actions both through the system and based on messages from the system, in some business context.

When stating that information systems are 1) "able to perform actions", they are thought of as agents acting on someone's behalf. IS action is always predefined and there is always a human actor responsible for it. Such responsibilities are important to identify and establish during systems engineering, and most certainly before implementation. Stating that information systems should "permit, promote and facilitate users to perform their actions" implies that systems should not only be usable and clear about its action potential but should also encourage users to take advantage of acting through the system. User actions could be performed 2) "trough the system". That is when a user performs actions with an IS as a vehicle for communication. On the contrary, 3) "based on messages from the system" refers to the case when an IS is used to create action possibilities. These three kinds of usage situations are referred to as "automatic", "interactive" and "consequential" respectively. The "degree" of actability possessed by a certain IS is always related to the particular business context. The business context includes actors pre-knowledge and skills regarding both the IS and the business task performed. Therefore IS actability is not a static property of an IS but depends on the social structures surrounding it.

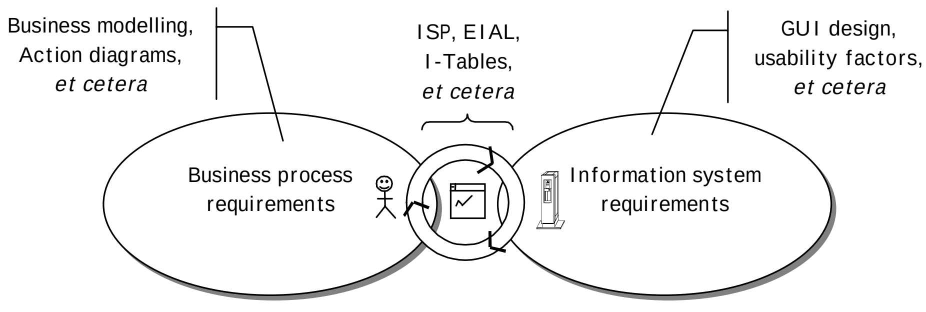

Figure 10: Actability — Bridging the requirements engineering gap.

By perceiving business and IS usage as action it is possible (and necessary) to design business processes and information systems as an integrated whole. We propose the use of Action Diagrams to describe action structures and to delineate interactive situation proposals (ISPs). The ISPs are then used as a basis for a deeper analysis of requirements concerning the design of interactive documents and action structures at a "lower" level. Such analysis leads to deeper understanding of the business and thus often reveals inconsistencies in previous assumptions. This way human-computer interaction design and implementation of different usability factors is both the result of business process modelling and an invaluable tool for its success.

The notions of ISP and EIAL, in conjunction with I-Tables, statecharts and conceptual modelling, to capture and describe actability requirements thus seem to lead to a bridge over the RE gap (see Figure 10).

References

Austin J L (1962). *How to do things with words*. Oxford University press.

Axelsson B, Easton G (Eds. 1992). *Industrial networks. A new view of reality*. Routlege, London.

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

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

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

Bubenko J A jr, Kirikova M (1999). Improving the Quality of Requirements Specifications by Enterprise Modelling. In Nilsson, Tollis & Nellborn (1999), *Perspectives on Business Modelling*. Springer Verlag, Heidelberg, Germany.

Card S K, Anderson A (1987). A multiple virtual-workspace interface to support task swithing. In *Proceedings of the Human Factors in Computing Systems (CHI'87)*, pp. 53-9. New York, ACM Press.

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

Davenport T H (1993). *Process innovation. Reengineering work through information technology*. Harvard Business School Press, Boston.

Denning PJ, Medina-Mora R (1995). Completing the loops. *Interfaces*, Vol. 25, No 3, pp. 42-57.

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

Goldkuhl G, Ågerfalk P J (1998). Action Within Information Systems: Outline of a Requirements Engineering Method. In Dubois, Pohl, Opdahl (eds.), *Proceedings of 4th Intl. Workshop on Requirements Engineering: Foundations of Software Quality (REFSQ’98)*. Pisa, Italy.

Goldkuhl G (1992). Contextual activity modelling of information systems. In *Proceedings of 3rd Intl. Working Conference on Dynamic Modelling of information systems*. Noordwijkerhout.

Goldkuhl G (1996). Generic business frameworks and action modelling. In Dignum & Dietz (1997), *Communication modelling - the Language Action Perspective*. Proceedings of the 2nd International workshop on Communication modelling, Computer science Reports, Eindhoven University of Tecnology.

Goldkuhl G (1998). *The Six Phases of Business Processes – Business Communication and the Exchange of Value*. Presented at Beyond Convergence: The 12th Biennial ITS Conference (ITS’98) in Stockholm. Jönköping Intl. Business School.

Graham I (1998). *Requirements engineering and rapid development: an object-oriented approach*. ACM Press. Addison Wesley.

Harel D (1987). Statecharts: A Visual Formalism for Complex Systems. *Science of Computer Programming*, 8 (1987) pages 231-274.

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

Horrocks I (1999). *Constructing the user interface with statecharts*. Addison-Wesley.

Jackson M, Twaddle G (1997). *Business Process Implementation: Building Workflow Systems*. ACM Press. Addison Wesley.

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

Lauesen S & Houman Y. (1998). Six Styles for Usability Requirements. In Dubois, Pohl & Opdahl (eds. 1998), *Proceedings of 4th Intl. Workshop on Requirements Engineering: Foundations of Software Quality (REFSQ’98)*. Pisa, Italy.

Ljungberg J, Holm P (1996). Speech Acts on Trial. *Scandinavian Journal of Information Systems*, Vol. 8, Nr. 1.

Löwgren J (1993). *Human-computer interaction - What every system developer should know*. Studentlitteratur. Lund. Sweden.

Löwgren J (1995). *Perspectives on Usability*. Dept. of Computer and Information Science, Linköping University, Sweden.

Mathiassen L, Seewaldt T, Stage J (1995). Prototyping and Specifying: Principles and Practices of a Mixed Approach. *Scandinavian Journal of Information Systems*, Vol. 7, Nr. 1.

Monk A, Wright P, Haber J, Davenport L (1993). *Improving your Human-Computer Interface*. New York: Prentice-Hall.

Nielsen J (1993). *Usability Engineering*. Academic Press, San Diego, CA.

Norman D A (1988). *The Psychology of Everyday Things*. Basic Books. New York.

Preece J, Rogers Y, Sharp H, Benyon D, Holland S, Carey T (1994). *Human-Computer Interaction*. Addison Wesley.

Searle J R (1969). *Speech acts. An essay in the philosophy of language*. Cambridge University Press, London.

Shneiderman B (1998). *Designing the user interface: strategies for effective human-computer-interaction*. Addison Wesley.

Winograd T, Flores F (1986). *Understanding computers and cognition: A new foundation for design*. Ablex, Norwood.

Ågerfalk P J, Goldkuhl G (1998). Elicitation and Analysis of Actability Requirements. In Fowler & Dawson (eds.), *Proceedings of 3rd Australian Conference on Requirements Engineering (ACRE'98)*. Geelong, Vic, Australia, Oct 26-27, 1998.