2007

Co-Evolution of Business Action Theories and Languages

Peter Rittgen

University College of Borås, peter.rittgen@hb.se

CO-EVOLUTION OF BUSINESS ACTION THEORIES AND LANGUAGES

Rittgen, Peter, University College of Borås, 501 90 Borås, Sweden, peter.rittgen@hb.se

Abstract

We suggest an incremental method for the co-evolution of theories of and modeling languages for business action. It consists of a cycle containing four steps: theory-based derivation of a generic modeling language, engineering of a language that is adapted to a specific modeling situation, the deployment of this language in a concrete analysis and/or design project, and the refinement of the theory with the help of action research. We apply this method to Business Action Theory and the language of the Situation-adaptable work and Information systems Modelling Method in the context of a business process-analysis project.

Keywords: Method engineering, Business Action Theory, Situation-adaptable work and Information systems Modelling Method, business process analysis.

1   INTRODUCTION

Theories of (social) action explain social behaviour in a general way with the help of a model that describes action patterns and a set of laws that determine which behaviour is allowed and which not. Examples of such theories are Actor Network Theory (Latour, 1987; Walsham, 1997), Structuration Theory (Giddens, 1984; Orlikowski, 1992), Activity Theory (Nardi, 1996; Vygotsky, 1962) and the language-action theories presented in the next section. The development (or rather evolution) of action theories is often associated with action research, a scientific method for creating and improving social theories (Baskerville & Wood-Harper, 1996; Susman & Evered, 1978). In it the researcher has a double role of being both the scientific observer and the active consultant who contributes to changes in the organization and thereby in the studied object. Although this implies a principal role conflict and loss of neutrality, it is typically argued that these disadvantages are outweighed by the increase in authenticity of the resulting theory with respect to the studied workpractice.

In action research we start with some preliminary theoretical knowledge that helps us in understanding the business situation of a particular case, e.g. a business process-analysis project. In the course of the project we will also encounter problematic situations that cannot be explained by the preliminary theory. In such a case the experiences from analysing the problematic situations can in turn contribute to an improved theory, e.g. by refining it, modifying it or adding to it. The improved theory is then the starting point for the next case study (round two) and so on. This cycle is repeated until theoretical saturation is reached (if at all).

Modeling languages are used to 'write' models. Models are vehicles to express views on perceived or constructed reality. As opposed to action theories, these models describe behaviour only with respect to a particular scenario without any claim to generality. The development of modeling languages also proceeds in cycles of language usage and language improvement (based on usage experience) in many cases, e.g. Event-driven Process Chains (Scheer, 1999), Unified Modeling Language (OMG, 2004; OMG, 2006) and the language of the Situation-adaptable work and Information systems Modelling Method (SIMM, explained later). It should be noted though that some approaches have a closer link between theory and language, for example, Dynamic Essential Modeling of Organizations (Dietz, 2006).

From the preceding paragraphs we see that the evolution of theory and language often proceeds in two similar but separate cycles. From a model-theoretic point of view this separation seems unnatural as both deal with the same social reality on different levels of abstraction. An integration of theory and language evolution is therefore in principal desirable. But to achieve it we must solve the issue of genericity vs. specificity. A modeling language containing concepts that are derived directly from an action theory is by nature situation-independent. If we want to develop models for a concrete business case, it is preferable to have a modeling language that is adapted to this situation. The missing link, i.e. the engineering of a case-specific language from a generic language, is provided by situational method engineering (Ralyté et al., 2003).

If we summarize the arguments above we arrive at a method for the co-evolution of theory and language that consists of the following steps (see also fig. 1):
- Deriving the concepts of a generic modeling language based on some action theory.
- Engineering a situation-specific language with the help of method engineering,
- Deploying the specific language in a modeling situation (analysis/design project),
- Learning from the language use to improve the underlying theory (action research).

In the following sections we exemplify this process of co-evolution. We use Business Action Theory (BAT) as the action theory and the language of SIMM as the modeling language.

We are not aware of other research that has been done on the co-evolution of modeling languages and theories but the more general question of the relation between languages and theories has been studied to some extent. The most well-known research are dealing with this issue is model theory (Manzano, 1999), which is concerned with set-theoretic interpretations of languages. But there the role of theory is to explain the language, whereas we are interested in theories that explain social action. A more closely related approach has been suggested by (Checkland, 2000). In LUMAS (Learning for a User

by a Methodology-informed Approach to a problem Situation) the user tailors a methodology to a real-world problem situation yielding a situation and user-specific approach. The LUMAS cycle is problem and usage-driven but it is not informed by theory. Our results suggest that the introduction of a theory cycle can enhance the stability of a generic modeling language over time, which forms an important part of a methodology.

2   BUSINESS ACTION THEORY

The literature on communicative action provides a broad spectrum of frameworks to describe business processes, e.g. Business Action Theory (Goldkuhl, 1996; Goldkuhl, 1998; Goldkuhl & Lind, 2004a), Dynamic Essential Modeling of Organizations (DEMO) (Dietz, 1999; Dietz & Habing, 2004), Action Workflow (Denning & Medina-Mora, 1995; Medina-Mora et al., 1992), Action-Based Modeling (Lehtinen & Lyytinen, 1986) and Conversation for Action (Winograd & Flores, 1986). Among these frameworks BAT can, in a certain sense, be seen as the most general because it does not commit the modeler to any specific language. On the one hand this is an advantage: The modeler can choose freely the language that is most appropriate in the actual application context. A possible choice would be that of the Situation-adaptable work and Information systems Modelling Method (Goldkuhl, 1996) as was suggested in the same paper that introduced BAT (ibid.).

But on the other hand the lack of a dedicated modeling language also represents a disadvantage because the choice of a particular language that was not tailored for BAT also implies that the modeler is not supported in applying BAT. As a theory represents stable and fundamental knowledge it makes sense to anchor a modeling language in it. These issues have been explored in several papers comparing BAT with DEMO (Reijswoud & Lind, 1998; Verharen, 1997) and Action Workflow (Goldkuhl, 1996; Verharen, 1997). There is an abundance of theories that are suitable for this purpose. In this paper we have chosen BAT as an example. Our primary purpose is to show how such anchoring can be done and not to argue for BAT as the "theory of choice".

Although the frameworks mentioned above are substantially different in many aspects they do largely agree on dividing a business process into phases. Among them BAT offers the most comprehensive phases: Business prerequisites phase, Exposure and contact search phase, Contact establishment and proposal phase, Contractual or commitment phase, Fulfilment phase and Completion or assessment phase. The other frameworks address only a part of these phases and/or they give different names to the phases and/or they subsume several phases under one heading. Action Workflow, for example, uses 4 phases which roughly correspond to phases 3 to 6 of BAT. DEMO introduces 3 phases, order (1), execution (2) and result (3), where the first subsumes phases 3 and 4 of BAT, while phases 2 and 3 correspond to phases 5 and 6 of BAT, respectively. As BAT offers the most comprehensive phases it is a reasonable starting point for business modeling. It has been introduced by (Goldkuhl, 1996) and was refined and adapted on the basis of further empirical evidence in (Goldkuhl, 1998; Goldkuhl & Lind, 2004a). One of the roots of BAT is Speech Act Theory (Austin, 1962; Searle, 1969) that views communication as action between (two) individuals, another one is the Theory of Communicative Action (Habermas, 1984), which puts action into a social context.

According to BAT business interaction involves two principal players, i.e. supplier and customer. At the core is the so-called business transaction that consists of the six phases which we have already mentioned. (Goldkuhl, 1996) identifies also a number of generic business actions that constitute the phases on the respective side of the transaction (i.e. supplier or customer). They are summarized in table 1.

Phase	Supplier	Customer
Prerequisites phase	Product/offer development	Identification of problems/needs
Exposure and contact search phase	Offer exposure	Contact search
Proposal phase	Offer	Inquiry
Commitment phase	Order confirmation	Order
Fulfilment phase	Delivery, Invoice, Receipt of payment	Receipt of delivery, Payment
Assessment phase	Acceptance, Claim	Acceptance, Claim

Table 1.   Generic Business Actions.

The business actions follow a certain execution logic but the whole transaction is by no means a linear, sequential procedure. In the proposal phase, for example, the supplier can make any number of offers concerning their products and/or services where each one will typically meet the customer's needs better than the preceding one. Likewise the customer can make a series of inquiries that usually become more and more "realistic". These loops terminate when offer and inquiry are sufficiently close to each other to reach an agreement whereupon we enter the contractual phase. In an ideal scenario this consists of the customer placing an order and the supplier confirming it. Both actions together constitute a contract the fulfilment of which is subject of the next phase. Here the supplier, again ideally, delivers the products/services and sends a corresponding invoice. The customer receives the delivery and makes the payment, which the supplier finally receives. In the completion phase each party decides whether they accept the receipt of the delivery/money or make a claim, i.e. request the fulfilment of that part of the contract they consider unfulfilled.

Orthogonal to the phases BAT offers another dimension, layers, that was introduced in (Lind & Goldkuhl, 2001). They extend and modify the layers originally suggested by (Weigand & Heuvel, 1998). Layers refer to the granularity of an action and in BAT they are, from fine grain to coarse grain: business act, action pair, exchange, business transaction and transaction group. A business act is a communicative act (speech act, e.g. placing an order) or a material act (e.g. performing a delivery). It is directed towards somebody with the aim of changing the world, i.e. the material world or the mental world (state of mind) of the addressee. An action pair is a pair of actions where the first one is a trigger (initiative) and the second a response. Actions can have a dual function so the response of one action pair can be the initiative of another.

3   DERIVING CONCEPTS OF A GENERIC LANGUAGE

A generic language for BAT has to take into account both dimensions of actions, phases and layers. The latter have been introduced in (Lind & Goldkuhl, 2001) and are shown in figure 1. A transaction group consists of a number of transactions. Each transaction is divided into exchanges. The exchanges correspond to the phases that have already been mentioned.
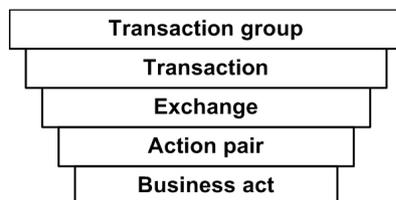

Figure 1.   The layers of BAT.

An exchange is made up of action pairs, each of which consists of two business acts. On the lowest layer a business act consists of one or more functions. The importance of these functions was already recognized in (Goldkuhl, 1996) where they were named mixed communicative actions. This suggests that a business act can be further divided into distinct, separate acts. But the mixed actions are rather different functions of the same act than different acts. We therefore prefer to call them the communicative and/or material functions of a business act. (Goldkuhl, 1996) gives the examples of

(commercial) offer and order. A commercial offer can be a single business act that has two communicative functions, that of requesting the potential customer to buy (i.e. to place an order), and that of committing the potential supplier to sell (i.e. to deliver) under certain conditions. If we apply a material and speech act analysis to the generic business actions we get the results shown in table 2.

Business Action	Material and/or communicative function
Offer exposure	*State* general offer
Contact search	*Express* interest
Inquiry	*Request* commercial offer + *Express* interest
Commercial offer	*Offer* delivery + *Request* order
Order	*Request* delivery + *Offer* payment
Order confirmation	*Promise* delivery
Delivery	*Transfer* merchandise/*Perform* service + *State* delivery
Invoice	*Request* payment + *State* contract fulfilment [supplier]
Receipt of delivery	*Accept* delivery + (*Accept* contract fulfilment [supplier])
Payment	*Transfer* money + *State* contract fulfilment [customer]
Receipt of payment	*Accept* payment + (*Accept* contract fulfilment [customer])
Acceptance	*Accept* contract fulfilment [supplier or customer]
Claim	*Request* contract fulfilment [supplier or customer]

Table 2.   Material and communicative functions of the genereic business actions.

These results show that a busines act typically has one or two functions. The communicative function is always present (even in the case of material acts) but there might also be another function that is either communicative or material. This is reflected in the model of figure 1. The generic business action "receipt of delivery or payment" can in some cases imply the acceptance of the contract fulfilment of supplier or customer, respectively. In other cases the acceptance is stated explicitly (i.e. separately in the assessment phase) or a claim is made (also in the assessment phase).

We are aware of the fact that such a list of generic actions and their functions can only serve as a recommendation that covers some typical or common situations. It is not meant to be a prescriptive template for all business interactions. The main purposes of it are rather as follows: First it should give an example of how material and speech act analysis can be used to identify material and communicative functions. Using that analysis in a different context might yield different actions and even different functions concerning the same actions. But the results can nevertheless be useful, and that is the second purpose, to find a set of recurring material and communicative functions that can be used as a pattern for a modeling language.

If we compile the identified material and communicative functions and sort them according to the illocutionary points introduced in (Searle, 1979), adding a column for material functions, we arrive at the structure show in table 3.

material	communicative				
	expressives	**declaratives**	**assertives**	**commissives**	**directives**
Transfer	Express	Accept	State	Promise	Request
Apply			Reply	Offer	Ask
Transform					
Perform					

Table 3.   Classification of material and communicative functions.

The material functions are transfering and object (i.e. moving it in space), applying an object as an instrument and transforming and object (i.e. changing some of its properties) possibly with the help of an instrument. The function "express" is used to show an emotion or an attitude (e.g. interest in a product). A directive is usually a request in a business context. A less formal and less compelling directive would be to ask a question. The reply is the corresponding assertive. There is another assertive, state, that carries a higher illocutionary force. It is a unilateral establishment of a fact, whereas the declarative "accept" is a confirmation of a stated fact, i.e. a mutual agreement on that fact.

An "accept" must therefore always be preceeded by a "state" because one party alone cannot declare agreement. The commissives are divided into promise and offer. The former is an unconditional commitment, the latter is subject to some conditions. If these conditions are fulfilled (typically by the other party) the offer becomes a promise. To avoid confusion of the communicative function "offer" with the same term in business we have called the latter a commercial offer. The function "perform" refers to a business act that is elementary at the current level of abstraction (i.e. with respect to the model under consideration) but a complex action on some lower, more detailed level.

The development of a set of material and communicative functions was motivated by (Lind & Goldkuhl, 2001) and (Goldkuhl, 1996). Both stress the importance of this issue (in the former paper it was called multi-functional business acts, in the latter mixed communicative actions). We agree with (Goldkuhl, 1996) that the illocutionary points of (Searle, 1979) are too coarse for business modeling and have therefore developed the set of functions in table 3 which is somewhat more elaborate and more adapted to business interaction. But nevertheless such a classification should be seen as a suggestion rather than a fixed template. It might require adaptation to a particular modeling scenario.

A classification of speech acts has also been done by (Reijswoud et al., 1999). They employed a purely theoretical method that consisted in viewing the one-dimensional classifications of Searle and Habermas, respectively, as two dimensions of a matrix. As a result they got the six speech acts question, answer, request, promise, state and accept. These are also found in table 3. Our classification can therefore be seen as an extension of that of (Reijswoud et al., 1999).

4 ENGINEERING A SPECIFIC LANGUAGE

The development of an entirely new language is a huge project. Such a project is only justified if the new language offers something substantially new. As we have mentioned earlier, there is already a number of methods that "implement" language-action concepts to some extent. We do therefore not propose a comprehensive new language but rather a set of elements that can, for example, be used to extend existing languages. The techniques for such an extension are offered by (situational) method engineering (Ralyté et al., 2003). The idea behind method engineering is to design methods in such a way that they fit the particular modeling situation. This can be done in different ways. One way is to extend an existing method. Another one is to create a new one from chunks of existing methods by performing method chunk selection and assembly. The third way is to construct a new method from scratch with the help of a suitable meta-model or paradigm. Using the first approach of method extension we enrich and refine the language of SIMM with the elements introduced in this section.

We propose that a business action language requires at least three element categories: actors, actions and (action) objects. As SIMM has the most elaborate concept of an action object, we borrow both the notion and the notation of an object from SIMM. Examples of information and material objects are shown in fig. 2 but SIMM offers many additional types. Actors are denoted by a rectangle containing the name of the actor as is common in many approaches. The actions themselves are divided into two categories according to the layer: business acts (layer 1) and the other layers. Actions on layers 2 to 5 are represented by a rounded rectangle with a double line. An additional classification symbol can be used to identify the particular layer: two intersecting circles for an action pair, two arrows pointing towards each other for an exchange, a "T" for a transaction and a "G" for a transaction group. For business acts in general we also use the rounded rectangle, for material acts the octagon. Both shapes have only one line to show that the act is elementary. The box can either contain the name of the business act or the respective material or communicative function where the function header is italicized. In the case of multiple functions the box can be divided into horizontal compartments, one for each function. If material and communicative functions are mixed we can also mix the respective shapes. Fig. 3 shows an overview of the notational elements for a business action language.
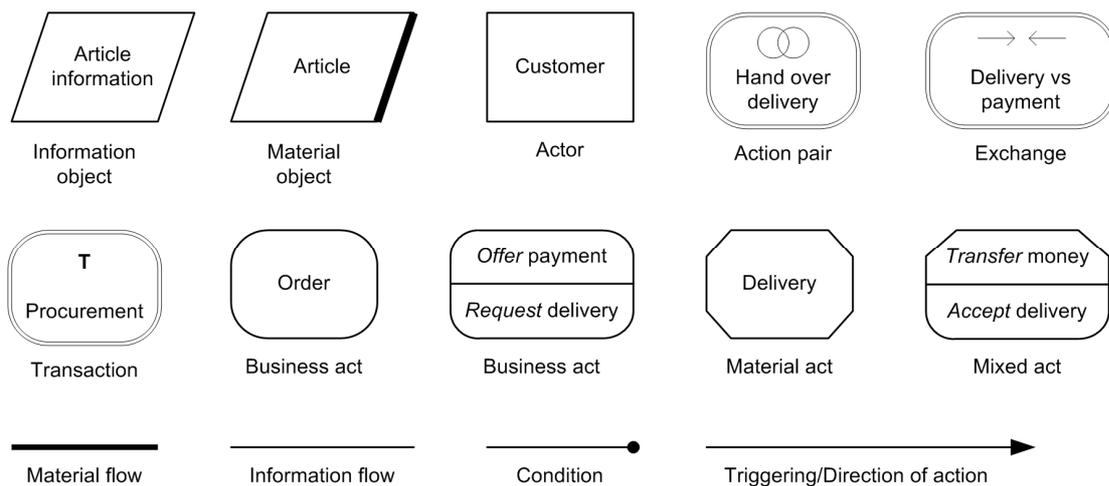
Figure 2.   *Notational elements for a business action language.*

Among the elements there are also four types of arcs. Two undirected arcs that represent an information flow (thin arc) or a material flow (thick arc). These have been borrowed from the SIMM Action Diagram where the direction of the flow is coincides with the drawing direction (from top to bottom). The condition arc allows us to show that one action is a condition for another action. The end with the black dot is attached to the latter action. The arrow serves two purposes. If it points from one action to another, the former triggers the latter. If it points from one actor to another, it represents an action that is directed from the first actor to the second. In this case the name of the action is written along the arrow. It can be accompanied by a symbol denoting the layer. For layers 2 to 5 we use the classification symbols introduced above. For communicative or material acts we use a small rounded rectangle (or circle) or a small octagon (or diamond), respectively. As an alternative to the arrow form of the action the boxed form of the action can be interlaced with the arrow.

As an example for the use of these elements let us consider an extension of the SIMM Action Diagram by the refined action concept presented above. We call the result an Action/Object Diagram. It can still express all the information in a SIMM Action Diagram but in addition we can also specify the action type (communicative or material), the layer and the communicative and material functions if we wish to do so (i.e. if the modeling situation requires that kind of information). A small example is shown in figure 3. It is taken from a business process that we have analyzed in the context of a project. The details of this project are given in the next section.
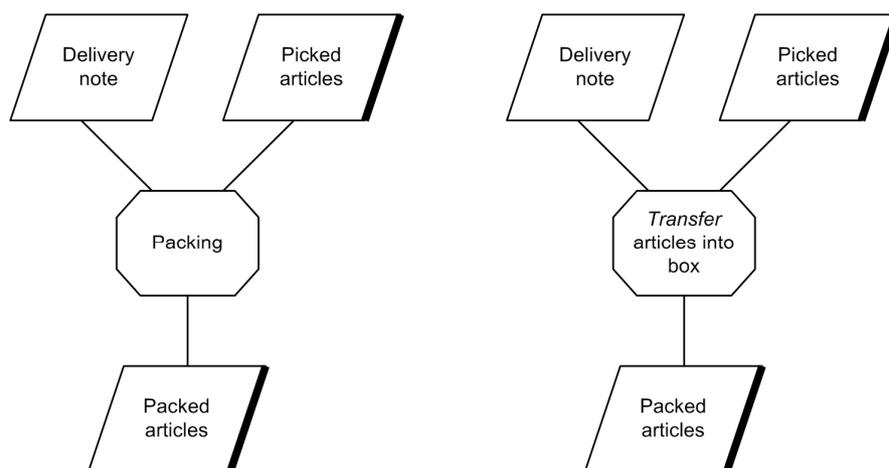

Figure 3.   *Example of an Action/Object Diagram.*

The Action/Object Diagram is an example of how the new language elements can be used to extend an existing language (that of SIMM). We use the language elements to define two diagram types that are suitable for analysing business processes with respect to how commitments are created and followed up. The next section elaborates this issue.

5   DEPLOYING THE SPECIFIC LANGUAGE IN BUSINESS-PROCESS ANALYSIS

Our work has been carried out in the context of a project that involved two companies that have a very close business relationship. One of them is the headquarters of a retail chain in the home textiles and decoration industry. The other is third-party logistics provider, let us call them LogPro, that performs all inbound and outbound logistics for the retailer. Our goal was to discover the major problems in their relationship and to suggest appropriate solutions. For this purpose we carried out an analysis of order processing and delivery that involved, apart from Headquarters and LogPro, also the shops of the chain. For this we first used the Interaction Diagram of SIMM. But then we discovered that we also need information on the type and level of an action so we enriched this diagram with the features introduced above. The resulting diagram is shown in fig. 4.

The process starts when Headquarters send an estimate regarding the capacity required for executing future orders. Such estimates are send six months, two months and two weeks in advance of the time of delivery. Shortly before that time the Shop can place different kinds of orders. A customer order is iniated by the Shop on behalf of a customer who wishes to buy an article that is not currently available in the Shop. The refill order is triggered by Headquarters whenever the Shop's stock is running low on articles of the basic assortment. Both actions are on the action-pair level because they require some kind of confirmation from the partner. The third type of order is called a distribution order. It is based on the budget that was negotiated before and the shop has to accept it as part of its franchise obligations. The distribution order is therefore only a single speech act that has a more informative character. The negotiation of the budget on the other hand is a bilateral process that is initiated by Headquarters but consists of an exchange of budget proposals.
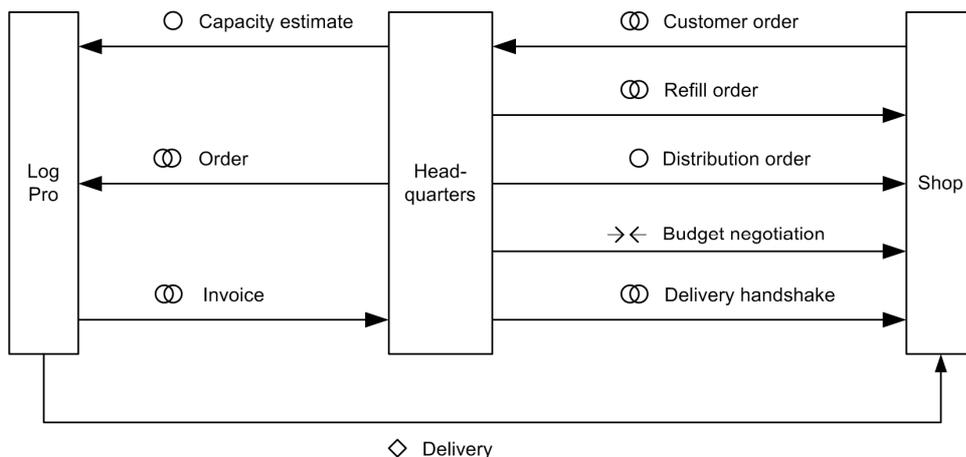

Figure 4.   Enriched Interaction Diagram.

Orders of all types are combined into one order by Headquarters and forwarded to LogPro. As a consequence LogPro will perform the delivery to the Shop. Headquarters inform the Shop about the upcoming delivery and receive a confirmation that is has arrived (delivery handshake). In regular intervals LogPro bill their services to Headquarters.

On the basis of this overview we developed detailed Interaction Diagrams for the interactions between Headquarters and Shop as well as between LogPro and Headquarters. The latter is shown in fig. 5. This diagram is on the business-act level, i.e. all actions in it are business acts. It shows that Headquarters send a capacity estimate first. On the day of delivery a pick file is transferred to LogPro that contains the order data. This is used by LogPro to pick the appropriate articles from the shelves and to pack them for delivery. As soon as the articles are on their way, LogPro reports the delivery to Headquarters. At the next billing occasion LogPro send an invoice and Headquarters makes the payment.
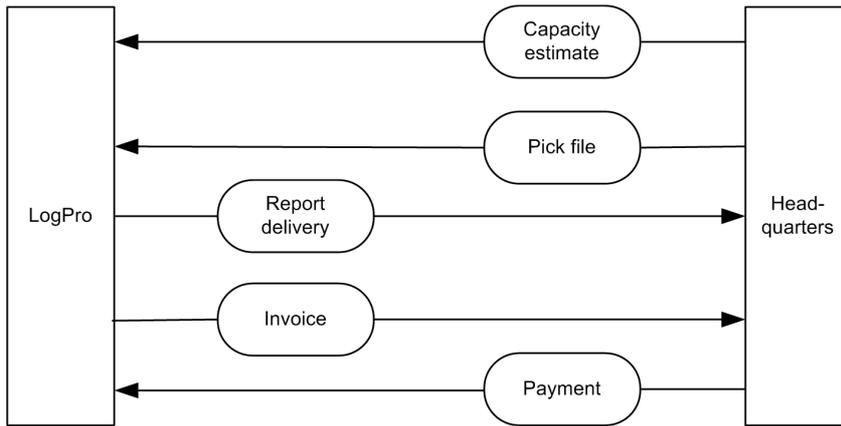
Figure 5.   Detailed Interaction Diagram.

For performing a commitment analysis we need more detailed information about how the actions are related to each other. This means that we have to exhibit the communicative and material functions that the actions have. These functions are the ones that lead to the establishment or fulfilment of commitments. When they have been made explicit we can show the conditional and causal relationships between the functions. This in turn helps us to uncover broken commitments. For this purpose we have created a new type af diagram, the Business Act Diagram. A diagram of this type for the relation between LogPro and Headquarters is shown in fig. 6.

Each actor box covers the actions that are performed by this actor. The capacity estimate is an action that implies both a request to provide this capacity and a promise to place an order that requires approximately the requested capacity. LogPro promise to provide this capacity. This offer is implicit (i.e. not communicated) because LogPro is required to provide the respective capacity by the terms of the frame contract. This triggers organizational measures to actually provide (offer) this capacity (e.g. staff planning). The provision of the capacity is a condition for the ability to perform the delivery that is triggered by the respective request from Headquarters which is a function of the order. The other function, request capacity, is is implicit and refers to the actual capacity required for handling the order (as opposed to the planned capacity of the first request). The delivery itself is both a physical transfer of the goods and a statement of the fulfilled commitment to deliver. The latter triggers acceptance of the delivery but only if the Shop has confirmed the arrival of the goods.
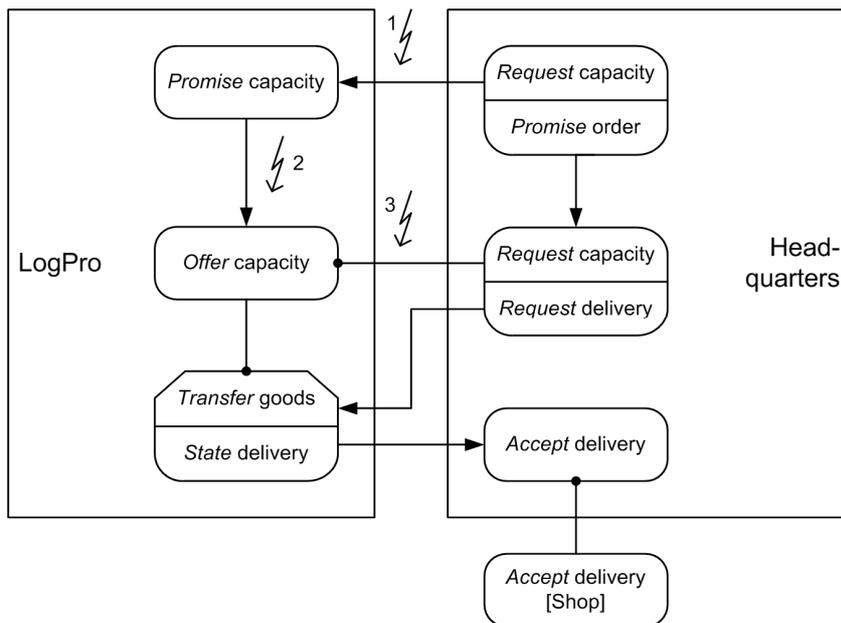

Figure 6.   Business Act Diagram.

1985

The Business Act Diagram has shown us that the pattern concerning the capacity is broken in three different places (see the flash symbols in fig. 6):

1. LogPro are unsure about the planned capacity that was requested as the request is in an unsuitable format (a huge, detailed Excel file that does not specify the weights and dimensions of each unit).

2. As a consequence LogPro can hardly plan for providing this capacity. But Headquarters will assume that the capacity is provided.

3. The actual capacity required for handling the order often deviates substantially from the planned capacity. This leads to higher costs and sometimes failure to meet the deadlines for delivery.

We have used this approach for other parts of the business process where we also succeeded in finding mistakes in commitment management. Among the problems we have identified this way are:

1. Broken patterns: One important characteristic of a business transaction is that each business act is related to another in a pattern of initiative and response. This means that the sequence of business acts needs to be followed in the sense that the pattern should not be broken. Going back to the empirical setting it can be identified that Headquarters supply estimates (as an initiative) without getting a response. There is thus a pattern of interaction when establishing the framework contract and another one when realizing the business transaction. The interaction pattern that glues framework contract and business process is thus broken. This has the effect that Headquarters cannot be sure of the capacity that will be available at the time of order and LogPro does not reserve the required capacity. The estimates made by Headquarters are therefore neither informative nor directive and hence do not imply mutual commitments. As a consequence, the contract should be specified in such a way that encourages the parties to keep the patterns intact.

2. Business rules: There are no rules that guide the interplay between the overall framework contract and the embedded business transactions. Such rules are necessary to regulate the details of interaction and to provide infrastructural support such as IT systems.

3. Indistinct communication structures: It was often unclear who communicates with whom regarding which issue. This led to excessive communication with the result that a considerable amount of time was wasted on solving minor everyday issues. This was only necessary because of the insufficient specification of routine procedures in the framework contract.

4. Lack of trust: Different interpretations of the framework contract by the parties led to misunderstandings and expectations that were not fulfilled. This caused in turn a lack of trust in the succeeding transactions which made the cooperation more and more difficult.

5. High transaction costs: Ad-hoc solutions to exceptional problems increased transaction costs. This was in part also due to missing business rules (issue 2) for situations that only occur occasionally but are not unforeseeable (e.g. out of stock). As such situations had not been provided for they required substantial effort at the time of their occurrence.

6   IMPROVING BUSINESS ACTION THEORIES

As a result of the deployment of the adapted SIMM language in the above mentioned project we arrived at a refinement of the BAT with respect to the structure of the layers (see fig. 7) with the exception of the transaction group layer. The transaction layer is divided into the exchanges (or phases) that have already been introduced. An exchange consists of two handover actions: One is directed from the supplier to the customer and the other vice versa. These handovers usually happen one ofter the other where the second happens in return for the first but the order is not predefined, i.e. in some cases the supplier hands over first and in others the customer. In certain cases, e.g. if the parties do not trust each other, the handovers can be near-simultaneous as for example in "delivery versus payment".

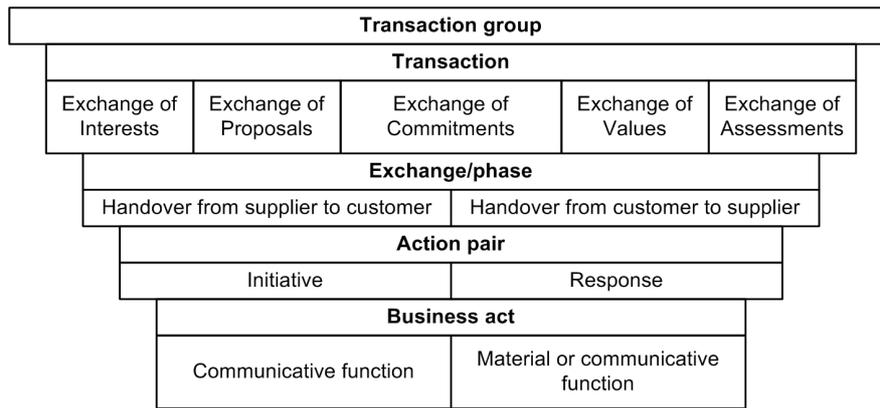| Transaction group				
Transaction				
Exchange of Interests	Exchange of Proposals	Exchange of Commitments	Exchange of Values	Exchange of Assessments
Exchange/phase				
Handover from supplier to customer		Handover from customer to supplier		
Action pair				
Initiative		Response		
Business act				
Communicative function		Material or communicative function		

Figure 7.   Refinement of the layers.

An action pair consists of two business acts, an initiative and a response. They have already been introduced as trigger and response in (Lind & Goldkuhl, 2001). On the lowest layer a business act consists of one or more functions as discussed earlier.

Based on the deployment of SIMM other refinements have been made e.g. of Business Action Theory (Goldkuhl, 1998; Goldkuhl & Lind, 2004a; Goldkuhl & Lind, 2004b; Goldkuhl & Melin, 2001; Lind & Goldkuhl, 1997; Lind & Goldkuhl, 2001; Lind & Goldkuhl, 2005) and of Theory of Practice (ToP) or Workpractice Theory (Goldkuhl & Röstlinger, 1999; Goldkuhl & Röstlinger, 2002; Goldkuhl & Röstlinger, 2006; Goldkuhl et al., 2001).

7   CONCLUSION

Action theories are stable frameworks for analysing business processes. They can guide the modeler in finding appropriate abstractions of the studied process and in relating different parts of the model to each other. But the support of the modeler can be strengthened by providing a modeling language that reflects the features of the theory. We have suggested a procedure to derive first a generic language that embodies the concepts of the theory and then a situation-specific language that is adapted to the case that we study. The latter is done with the help of techniques from situational method engineering. The deployment of the adapted language can then support us in understanding a business situation and in detecting and analysing problems in a more structured fashion than a general purpose language. But at the same time the modeler/action researcher discovers also the shortcomings in its use that can point to shortcomings in the underlying theory. An analysis of them will ultimately lead to an improved theory, thus completing the circle of theory and language co-evolution.

We have applied this general procedure to BAT and the SIMM modeling language. In this process we adapted SIMM to business analysis with a focus on commitment management. This helped us both with identifying broken commitments and with improving BAT by refining the layers. We have thus completed the whole cycle and shown that such a procedure is feasible as well as useful. Additional research is required, though, to confirm these initial results and to deepen our knowledge about the co-evolution of theories and languages beyond the case and the context we have chosen.

So far we have studied our approach only in the limited context of a specific language and theory. Both BAT and SIMM are well aligned with each other. This is not so much happenstance or the result of a careful choice of two fitting candidates but rather the consequence of the co-evolution process they have undergone. The approach we describe is therefore not suitable for any combination of theory and modeling language per se. There is a good chance though that an arbitrary modeling language can be informed by a related theory and vice versa. This is because the general procedure makes no direct reference to particular properties of the language or the theory. As a result, a modeling language and a theory that are related but not aligned with each other will "grow together" after a number of evolution cycles.

References

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

Baskerville, R. L. and Wood-Harper, A. T. (1996) A critical perspective on action research as a method for information systems research. *Journal of Information Technology* 11 (3), 235-246.

Checkland, P. (2000) Soft Systems Methodology: A Thirty Year Retrospective. Systems Research and Behavioural Science 17(S1): S11-S58.

Denning, P. J. and Medina-Mora, R. (1995) Completing the loops. *Interfaces* 25 (3), 42-57.

Dietz, J. L. G. (1999) Understanding and modeling business processes with demo. In *Proceedings of the 18th international conference on conceptual modeling er '99* (Akoka, J. and Bouzeghoub, M. and Comyn-Wattiau, I. and Métais, E., Eds), pp 188-202, Springer, Berlin.

Dietz, J. L. G. (2006) *Enterprise Ontology: Theory and Methodology*. Springer, Berlin.

Dietz, J. L. G. and Habing, N. (2004) The notion of business process revisited. In *Proceedings of the otm confederated international conferences, coopis, doa, and odbase* (Meersman, R. and Tari, Z., Eds), pp 85-100, Springer, Berlin.

Giddens, A. (1984) *The constitution of society. Outline of the theory of structuration.* Polity Press, Cambridge.

Goldkuhl, G. (1996) Generic business frameworks and action modelling. In *Communication modeling - the language/action perspective, proceedings of the first international workshop on communication modeling* (Dignum, F. and Dietz, J. and Verharen, E. and Weigand, H., Eds), Springer, Berlin.

Goldkuhl, G. (1998) The six phases of business processes - business communication and the exchange of value. In *12th biennial ITS conference "Beyond convergence" (ITS´98), Stockholm.*

Goldkuhl, G. (2005) Socio-instrumental pragmatism: A theoretical synthesis for pragmatic conceptualisation in information systems. In *3rd International Conference on Action in Language, Organisations and Information Systems (ALOIS)*, University of Limerick.

Goldkuhl, G. and Lind, M. (2004a) Developing e-interactions – a framework for business capabilities and exchanges. In *12th European Conference on Information Systems*, Turku, Finland, June 14-16, 2004.

Goldkuhl, G. and Lind, M. (2004b) The generics of business interaction - emphasizing dynamic features through the bat model. In *9th International Working Conference on the Language-Action Perspective on Communication Modelling*, New Jersey, Rutgers University.

Goldkuhl, G. and Melin, U. (2001) Relationship management vs business transactions: Business interaction as design of business interaction. In *10th International Annual IPSERA Conference*, Jönköping International Business School.

Goldkuhl, G. and Röstlinger, A. (1999) Expanding the scope: From language action to generic practice. In *4th Int Workshop on the Language Action Perspective (LAP99)*, Copenhagen.

Goldkuhl, G. and Röstlinger, A. (2002) The practices of knowledge – investigating functions and sources. In *3rd European Conference on Knowledge Management (3ECKM)*, Dublin.

Goldkuhl, G. and Röstlinger, A. (2006) Context in focus: Transaction and infrastructure in workpractices. In *4th Intl Conference on Action in Language, Organisations and Information Systems (ALOIS-2006)*, Borås.

Goldkuhl, G., Röstlinger, A. and Braf, E. (2001) Organisations as practice systems – integrating knowledge, signs, artefacts and action. In *Organisational Semiotics, IFIP 8.1 Conference*, Montreal.

Habermas, J. (1984) *The theory of communicative action 1 - reason and the rationalization of society.* Beacon Press, Boston.

Latour, B. (1987) *Science in action: How to follow scientists and engineers through society.* Open University Press, Milton Keynes.

Lehtinen, E. and Lyytinen, K. (1986) An action based model of information systems. *Information Systems* 11 (4), 299-317.

Lind, M. and Goldkuhl, G. (1997) Reconstruction of different business processes - a theory and method driven analysis. In *2nd International Workshop on Language/Action Perspective (LAP97)*, Eindhoven University of Technology, The Netherlands.

Lind, M. and Goldkuhl, G. (2001) Generic layered patterns for business modelling. In *Sixth International Workshop on the Language-Action Perspective on Communication Modelling (LAP 2001)*, Montreal, Canada, July 21-22, 2001.

Lind, M. and Goldkuhl, G. (2005) Designing business process variants. In *Business Process Design Workshop at the Third International Conference on Business Process Management*, Nancy, France, September 5-8, 2005.

Manzano, M. (1999) *Model Theory*. Oxford University Press, Oxford.

Medina-Mora, R., Winograd, T., Flores, R. and Flores, F. (1992) The action workflow approach to workflow management technology. In *Proceedings of the conference on computer-supported cooperative work cscw'92* (Turner, J. and Kraut, R., Eds), pp 281-288, ACM, New York.

Nardi, B. A. (1996) *Context and consciousness. Activity theory and human-computer interaction.* MIT Press, Cambridge.

Omg (2004) Uml 2.0 superstructure specification. OMG, Needham; MA.

Omg (2006) Unified modeling language: Infrastructure. OMG, Needham, MA.

Orlikowski, W. J. (1992) The duality of technology: Rethinking the concept of technology in organizations. *Organization Science* 3 (3), 398-429.

Ralyté, J., Deneckère, R. and Rolland, C. (2003) Towards a generic model for situational method engineering. In *Proceedings of 15th international conference on advanced information systems engineering (caise 2003)* (Eder, J. and Missikoff, M., Eds), pp 95-110, Springer, Berlin.

Reijswoud, V. E. V. and Lind, M. (1998) Comparing two business modelling approaches in the language action perspective. In *Language Action Perspective (LAP´98)*, Stockholm.

Reijswoud, V. E. V., Mulder, H. B. F. and Dietz, J. L. G. (1999) Communicative action-based business process and information systems modelling with demo. *Information Systems Journal* 9 (2), 117-138.

Scheer, A.-W. (1999) *Aris - business process modeling.* Springer, Berlin.

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

Searle, J. R. (1979) *Expression and meaning. Studies in the theory of speech acts.* Cambridge University Press, London.

Susman, G. I. and Evered, R. D. (1978) An assessment of the scientific merits of action research. *Administrative Science Quarterly* 23 (4), 582-603.

Verharen, E. (1997) A language-action perspective on the design of cooperative information agents. Katholieke Universiteit Brabant, Tilburg.

Vygotsky, L. S. (1962) *Thought and language.* MIT Press, Cambridge.

Walsham, G. (1997) Actor-network theory: Current status and future prospects. In *Information systems and qualitative research* (Lee, A. S. and Liebenau, J. and Degross, J. I., Eds), Chapman & Hall, London.

Weigand, H. and Heuvel, W. J. V. D. (1998) Meta-patterns for electronic commerce transactions based on flbc. In *Hawaii International Conference on System Sciences (HICSS '98)*, IEEE Press.

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