Behaviour Alignment as a Mechanism for Anticipatory Agent Interaction

Gerben G. Meyer[1], Nick B. Szirbik[2]

[1]The Agent Laboratory, Faculty of Management and Organization, University of Groningen, Nijenborgh 4, 9747 AG Groningen, The Netherlands, +31 (0)50 3638496
gerbenm@gmx.net
[2]Dept. Information Systems, Faculty of Management and Organization, University of Groningen, Landleven 5, PB 800, 9700 AV Groningen, +31 (0)50 363 {8125 / 8497}
n.b.szirbik@rug.nl

Abstract. In this paper, we present a formalism to define agents' behaviours (as exhibited in agent to agent interactions), by an extension of Petri Nets, and show how behaviours of different agents can be aligned using specific alignment policies. We explain why these agents are anticipatory, and the link between Business Information Systems and anticipatory systems is elaborated. A mechanism is proposed for automatic choosing of an alignment policy by the agent, in order to make the system more reliable, and to reduce the necessary human intervention. Due to the preliminary nature of this work, future directions of research are pointed out.

1 Introduction

In the anticipatory system research community, the agent based computing area is considered a promising one. However, there is little interest yet in applying the anticipatory agent concept in a real setting. Seminal work of Davidsson, Astor and Ekdahl [4], pointed out that agents can be characterised as agents when their acting can be described by a social theory. We argue in this paper that business organisation are in fact anticipatory systems themselves. Especially when these use an information system (usually called BIS - Business Information System). Our research group [14] is investigating novel agent-based architectures and development frameworks [10]. We recognise the importance of the anticipatory system concept in this context and position our models of organisations in the initial definition of Rosen ([11], page 339):

> "We tentatively defined the concept of an anticipatory system: a system containing a predictive model of itself and/or of its environment, which allows it to change state at an instant in accord with the models prediction to a latter instant."

In this paper, which should be seen as a position paper, presenting preliminary research, we investigate how the anticipatory ability of a single agent can be expressed as an interaction belief and also the way this belief can be changed. We will describe a policy for alignment that can be applied when the interaction beliefs of two or more interacting agents are not matching. We will introduce an extension of Petri Nets to capture the interaction beliefs and also a mechanism to choose the appropriate policy that adapts the beliefs from one agent perspective. From the anticipatory systems perspective, this research can enable predictive agent model execution (agent-based simulation of organisational models) to be more reliable and necessitate less human intervention in terms of alignment.

This paper is organised as follows: the rest of the introduction makes the link between BIS and anticipatory systems and gives the motivation for this line of research, section 2 deals with the formalisation of interaction beliefs of agents as Behaviour Nets, section 3 shows how Behaviour Nets of different agents can be aligned by using a specific policy. The paper concludes by discussing the standing issues and questions, pointing towards future research lines.

1.1 Motivation

Business information systems have evolved from a data centric perspective to a process centric perspective. The role of these systems is to support human activity in a business organisation. They support at a basic level information storage and retrieval, information flows and information processing. At a higher level they support human decision making. Depending on the time horizon, the decision can be related to operational management (day-to-day activities), tactical planning (week/month projections), strategic decisions (month/year projections), and even policy implementation (very long term).

The move from data centric to process centric systems did not change the centralistic nature of these systems. The way the system is designed and used ascribes to the notion that there exists an external observer that is able to investigate and understand the processes within the organisation. These processes can be identified in a semantic sense and modelled in a syntactic sense, that is, models of the processes can be described in a (semi) formal language. These models can be used to implement systems that support the actors that execute the process in the organisation.

The actors that are executing the organisations' processes have only local, often conflicting views. If the system is to be designed and implemented by allowing local and different models of the participating actors, a distributed, agent-oriented approach is more suitable. Agent-based modelling and agent-software engineering have been very popular in the last decade and paved new avenues for the development of the business systems of tomorrow. However, as correctly pointed out by Ekdahl [5], the lack of a strict definition of an agent and a clear view about what exactly agent software engineering is, many development processes tend to be in name agent-oriented, in reality, they can be just classified as purely reactive systems. He also states:

> "More sophisticated anticipatory systems are those which also contain its own model, are able to change model and to maintain several models,

> which implies that such systems are able to make hypotheses and also that they can comprehend what is good and bad."

One can infer from this statement that true agent systems are only those that have a clear anticipatory ability, both at the level of the individual agents themselves, and also at the whole multi-agent system. The ability to reason about a plan in an organisation is usually realised via humans. If one tries to simulate a planning organisation, a typical barrier is the evaluation of the plans. Such simulation tend to become interactive games, where the "players" (i.e. the expert planners) are becoming decision makers that select the "best" plan. Various plan selection mechanisms can be enacted, but these are usually just models of the behaviour of the players. In a monolithic, centralistic ERP system for example, this will be implemented as a single utility function that characterises the whole organisation, which makes explicit the criteria against which a prospective plan is checked. In reality, many expert players are co-operating with the system to adjust and decide for the best plan. The overall behaviour of the organisation (in terms of planning) is just emerging as a combined behaviour of the experts and the system that supports them.

This observation leads to the natural conclusion that it is better to enact decision support structures that mimic the distributed nature of this environment. Attempts to model and implement agent-oriented support for planning and other business processes are still in their infancy, but even simple implementations of crude multi-agent architectures show a higher degree of adaptiveness and flexibility.

1.2 Our approach towards anticipatory agents

Our research team is developing agents via simulation-games [14], where the behaviour of the software agents is captured from the expert players. These human experts can describe their intended behaviour, in terms of activities and local goals, but also can describe the behaviour they expect from the other agents in the game. These behaviours can be simplified and formally described. From a local perspective the *intended behaviour* of self and the *expected behaviour* of others can be seen as a specific *interaction belief* of that agent. The organisations' processes can be viewed as a set of running interactions. Each interaction is executed by the agents that play the roles that define the interaction and the execution depends on the (local) interaction beliefs. If the agents have consistent beliefs, a coherent execution of the interaction will take place. In an environment where human agents are playing the roles, slight (or even severe) misalignment of these behaviours can be solved by the capacity of the humans to adapt to misunderstandings and information mismatch.

Agents (as humans) develop over time a large base of interaction beliefs, which allow them to cope with a wide range of interaction situations. This is why the organisational processes can be carried out in most contexts and exceptional situations. In these, the monolithic and centralistic support of BPM becomes a problem in itself, needing roll-back procedures and "backdoor" interventions. When using an agent-oriented approach, in order to solve the exceptions that occur but have no resolution beliefs implemented in the software agents, a "escape/intervention" [10] mechanism can be used. Each time an agent cannot find a local solution for a mismatch during an interaction, it can defer control to a higher authority (higher level agent, typically a human). Therefore, such a system will never block, supporting the humans up to the levels it has been programmed to do, but leaving the humans to intervene when the situation is too complex for them to solve.

Interaction beliefs are local anticipatory models. These describe future possible states in a specific interaction from a local perspective of an agent. In an organisation, an agent can play various roles by using her "experience" (interaction beliefs that have proved successful in the past), but can also build new ones, depending on the context. Continuous enactment of interaction leads to whole process enactment. In a software multi-agent system, if the captured behaviours are not matching in a given context, the agents will revert to humans. Of course, this can decrease the performance of the system - in terms of support and/or automation - to unacceptable levels. Software agents should be able also to adjust their behaviour in an anticipatory way. There are two ways to tackle behaviour mismatches:

- there is a superior of the two agents that can align and impose a common interaction behaviour that is sound, by having full access to the interaction beliefs of the agents. This can happen before the interaction starts
- each agent is trying to align her behaviour on-the-fly, having only local information

In the next two sections, we describe a method to implement the second choice, by using a representation of the behaviour in terms of Behaviour Nets and a mechanism based on "alignment policies". We considered that the first choice is "less anticipatory", in the sense that only if viewed from a larger perspective (the system is formed by the participating agents, plus the superior agent - we call this a deus ex machina) becomes a system that investigates a potential scenario for the future. In the "on the fly" mechanism, the anticipatory system is the individual agent who tries to align its behaviour, based on the limited information she has about the interaction execution.

2 Behaviour Nets

Petri Nets are a class of modeling tools, which originate from the work of Petri [9]. Petri Nets have a well defined mathematical foundation, but also a well understandable graphical notation [12]. Because of the graphical notation, Petri Nets are powerful design tools, which can be used for communication between the people who are engaged in the design process. On the other hand, because of the mathematical foundation, mathematical models of the behaviour of the system can be set up. The mathematical formalism also allows *validation* of the Petri Net by various analysis techniques.

The classical Petri Net is a bipartite graph, with two kind of nodes, *places* and *transitions*, and directed connections between these nodes called *arcs*. A connection between two nodes of the same type is not allowed. A transition is *enabled*, if every input place contains at least one token. An enabled transition may fire, which will change the current marking of the Petri Net into a new marking. Firing a transition will *consume* one token from each of its input places, and *produce* one token in each of its output places.

2.1 Definition of Behaviour Nets

In the following, the formal definition of Behaviour Nets is given, which is a Petri Net extension, based on Workflow Nets [1], Self-Adaptive Recovery Nets [7] and Coloured Petri Nets [8]. An example of such a Behaviour Net can be seen figure 2 (a).

Definition 1. *Definition of Behaviour Nets*

A Behaviour Net is a tuple $BN = (\Sigma, P, Pm, T, Fi, Fo, i, o, L, D, G, B)$ where:

- Σ is a set of data types, also called colour sets
- P is a finite set of places
- Pm is a finite set of message places (such that $P \cap Pm = \emptyset$)
- T is a finite set of transitions
- $Fi \subseteq ((P \cup Pm) \times T)$ is a finite set of directed incoming arcs
- $Fo \subseteq (T \times (P \cup Pm))$ is a finite set of directed outgoing arcs (such that $Fi \cap Fo = \emptyset$)
- i is the input place of the behaviour with $\bullet i = \emptyset$ and $i \in P$
- o is the output place of the behaviour with $o\bullet = \emptyset$ and $o \in P$
- $L : (P \cup Pm \cup T) \rightarrow A$ is the labeling function where A is a set of labels
- $D : Pm \rightarrow \Sigma$ denotes which data type the message place may contain
- G is a guard function which is defined from Fi into expressions which must evaluate to a boolean value
- B is a binding function defined from T into a set of bindings b, which binds values (or colours) to the variables of the tokens

The set of types Σ defines the data types tokens can be, and which can be used in guard and binding functions. A data type can be arbitrarily complex, it can be for example a string, an integer, a list of integers, or combinations of variable types.

The places P and Pm and the transitions T are the nodes of the Behaviour Net. All these three sets should be finite. The extension of classical Petri Nets is the addition of the set Pm which are nodes for sending and receiving messages during an interaction. Such a message place is either a place for receiving or for sending messages, it cannot be both.

Fi and Fo are the sets of directed arcs, connecting the nodes with each other. An arc can only be from a place to a transition, or from a transition to a place. By requiring the sets of arcs to be finite, technical problems are avoided, such as the possibility of having a infinite number of arcs between two nodes.

Execution a behaviour is part of an interaction process, the behaviour is created when the interaction starts, and deleted when the interaction is completed. For this reason, the Behaviour Net also has to have one input and one output node, because the Behaviour Net initially has one token in the input place when the interaction starts, and can be deleted when there is a token in the output place.

With function L, a label can be assigned to every node. This has no mathematical of formal purpose, but makes the Behaviour Net better understandable in the graphical representation.

Function D denotes which message place may contain which data type. This is useful for determining which message place an incoming message has to be placed on. Because the two (or more) behaviours in an interaction are distributively executed, message places of both behaviours cannot be connected directly with each other, as the behaviours do not have to be aligned.

Function G is the guard function, which expresses what the content of a token has to be, to let the transition consume the token from the place. Function G is only defined for Fi, because it makes no sense to put constraints on outgoing edges of transitions. In other words, this function defines the preconditions of the transitions.

Transitions can change the content of a token. Binding function B defines per transition, what the content of the tokens produced by the transition will be. Bindings are often written as for example: $(T1, < x = p, i = 2 >)$, which means that transition $T1$ will bind value p to x and value 2 to i. The values assigned to the variables of the token (which data type must be in Σ) can be constants, but can also be values of the incoming token, or values from the knowledge- or belief-base of the agent.

2.2 Operations

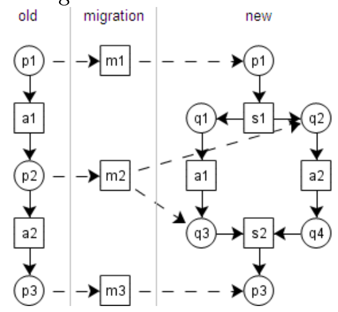
In Behaviour Nets, there are some *primitive* operations for modifying the net structure, such as adding and deleting places, transitions, arcs and tokens. Besides the primitive operations, there is a set of more *advanced* operations, which also preserve local soundness. By preserving local soundness we mean that after applying the operation, an execution of the behaviour will still terminate properly, if the behaviour also terminated properly before the operation. The message places Pm are not taken into account when determining local soundness. Local soundness refers to a sound behaviour, to make the distinction with a sound interaction, which will be referred to as global soundness. More information about soundness can be found in [2]. The used set of advanced operations are:

- `division` and `aggregation`*, which divides one transition into two sequential transitions, and vice versa,

- parallelization and sequentialization*, which puts two sequential transitions in parallel, and vice versa,
- specialization and generalization, which divides one transition into two mutual exclusive specializations, and vice versa,
- iteration and noIteration, which replaces a transition with an iteration over a transition, and vice versa,
- receiveMessage and notReceiveMessage, which adds or deletes an incoming message place,
- sendMessage and notSendMessage, which adds or deletes an outgoing message place.

For some of the operations, marked with *, is it not always clear how they can be applied on-the-fly, because of the dynamic change problem [3]. For example, sequentialization, as mentioned above, cannot be applied for every token marking, as it is not always clear on which places the tokens from the old behaviour should be placed, when migrating to the new behaviour. For modeling the migrations the approach of Ellis et al. [6] is used. By modeling a behaviour change as a Petri net, it can be exactly defined how to migrate the tokens from the old behaviour to the new behaviour. Note that advanced operations can also be described using the primitive operations. For the receiveMessage, notReceiveMessage, sendMessage and notSendMessage, nothing needs to be migrated, as there is no change in the places, except for the message place, which initially don't contain a token. In figure 1 can be seen how the migration for the operation parallelization can be modelled.

Fig. 1. Migration of old to new behaviour



3 Aligning Behaviours

Before two agents start an interaction, they will both individually choose a behaviour they are going to execute, based on what they are expecting of the interaction. An interaction however will not terminate, if the behaviours of the agents interacting are not matching. To overcome this problem, agents are able to change their behaviour on-the-fly, i.e. during the interaction. Alignment policies are used by agents to change their behaviour on-the-fly.

3.1 Alignment policies

An alignment policy is a set of primitive or advanced operations. In our approach, an agent has a set of policies in his knowledge-base from which she can choose when an interaction for example has deadlocked, i.e. when there is no progression anymore in the execution of the behaviour. How an agent will choose an alignment policy (or if she will choose one at all) depends on different factors. The factors discussed next are: kind of problem, beliefs about the agent interacting with, and the willingness to change it's own behaviour.

Kind of problem Most of the time, a problem will occur, when the agent is not receiving the message she is expecting. It can be that the agent did not receive a message at all, or received a different type of message than expected. If she did receive a message, the type of the received message and other factors of the kind of the problem can be used as attributes for selecting the proper alignment policy.

Beliefs about the agent interacting with Beliefs about the other agent can be of great importance when choosing an alignment policy. When for example the agent completely trusts the other agent, she might be willing to make more "sacrifices" in changing her behaviour than when she distrusts the other agent.

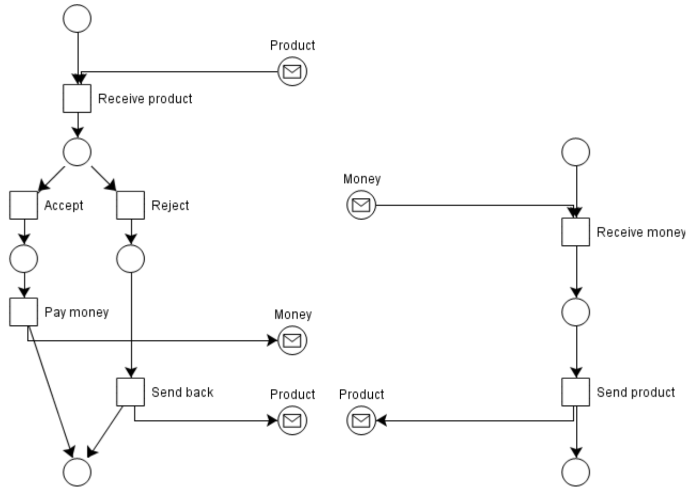
Willingness to change behaviour When an agent has very advanced and fine-tuned behaviours, it is not smart to radically change the behaviours because of one exceptional interaction. On the other hand, when the behaviour of the agent is still very primitive, changing it a lot could be a good thing to do. So when an agent gets "older", and the behaviours are based on more experience, the willingness to change her behaviour will decrease. This approach can be compared with the way humans learn, or with the decreasing of the learning rate over time when training a neural network.

As all this still needs research for what is the best way is to make the decision, a possibility would be to use a heuristic (for example machine learning techniques like a decision tree, neural network or a genetic algorithm), build on experiences of the use of the alignment policies in previous interactions. A new agent won't have any alignment policies, or experience applying them. When an agent does not know how to handle a certain problem, it can go into escape mode, to learn new ways to overcome her lack of experience. More information about the concept of escape mode can be found in [10].

3.2 Example - Proof of concept

As an example how these alignment policies could work, we give a small example, as a proof of concept. In this example, as shown in figure 2, the buyer and the seller already agreed on the product the buyer wants to buy, but as seen in the figure, they have different ideas of how the delivery and the payment should go. For the sake of the example, we assume that the behaviour of the buyer is very advanced, and thus has no willingness to change her behaviour. On the other side, the seller's behaviour is still primitive and unexperienced, so we are looking at the problem how the seller can align her behaviour with the buyer, assuming that the seller has trust in the buyer.

Fig. 2. Behaviours of buyer and seller

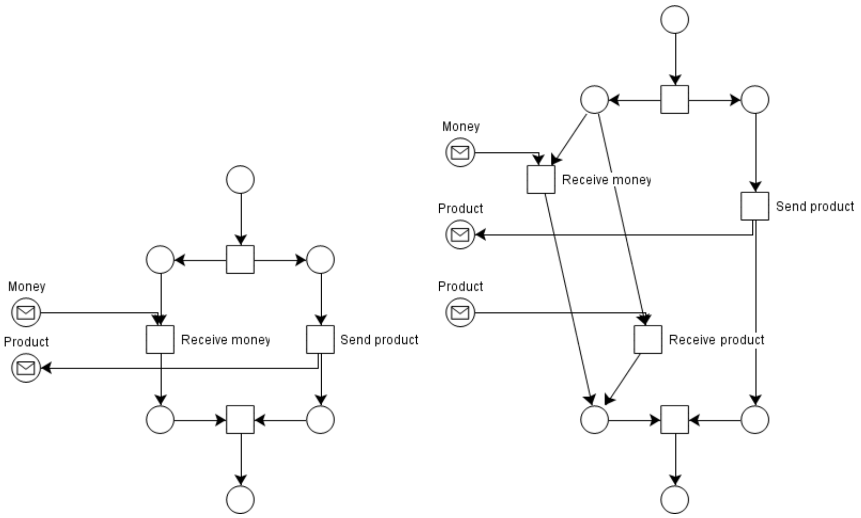


(a) Buyer (b) Seller

When the interaction starts, it immediately deadlocks; the buyer is waiting for the product, and the seller is waiting for the money. A way to overcome this problem would be for the seller to send the product and wait for the money in parallel. So, by using an alignment policy based on the operation `parallelization` the behaviour of the seller changes to the behaviour as seen in figure 3 (a), and the interaction can continue. However, if the buyer rejects the product, and sends it back, the seller still doesn't have the appropriate behaviour to handle this, because the seller is waiting for the money. In case the seller receives the product back, but when she is expecting the money, the seller could use an alignment policy based on the operation `specialization` to overcome this problem, which divides the receive money transition into two separate transitions, *receive money* and *receive product*. The resulting behaviour can be seen in figure 3 (b). The behaviours of the buyer (figure 2 (a)) and the behaviour of the seller (figure 3 (b)) are now aligned, and thus matching.

An extension of this example showing a more complex interaction concerning a typical business situation is shown in [13].

Fig. 3. Adapted behaviour of seller



(a) First adaptation

(b) Second adaptation

4 Discussion and Future Work

This research is preliminary. As far as we know, it is the first attempt to apply this kind of discrete mathematics to anticipatory agents. Although there are some approaches that apply Petri Nets to model agent interaction, these are mainly concerned with a centralistic view. However, we are taking a distributed approach. This approach has the potential to appeal to two research communities: the one oriented towards Business Information Systems development (who apply Petri Net like modelling to BPM and ERP), and also to the growing anticipatory agent community. Some researchers have pointed out that the models used for BIS analysis and design are in fact executable models of the organisation they support. Apparently, the inclusion of a executable model of the organisation in the organisation itself (seen as a system), makes the whole an anticipatory system. Obviously, organisations that use a BIS increase their anticipatory ability. Unfortunately, there is no evidence that the current development of BISs is done with explicit anticipatory ability in mind.

Our strong belief is that agent-oriented BIS that support the business processes of the organisation (in terms of interaction support), due to the anticipatory ability of the individual agents, lead to an emergent behaviour of the whole system that has a anticipatory nature. Of course, such a statement has to be proven empirically and theoretically. An intuition is that simulation - currently intended for development purposes - can have an important role in the anticipatory architecture of an agent-enabled BIS in an organisation. If the executable agent-based model of the organisation can perform simulations itself that start from the present state as perceived in the organisation, this model can predict future states. The results of these predictive simulations can be used to influence via an effector sub-system the current state of the organisation.

Currently, the idea about development simulations is that these are in fact games, where expert players interact with the simulated agents, via the escape/intervention mechanism. An escape is triggered when an agent cannot perform a certain act, and an intervention is when the human supervisor decides that the course of action is not proper. After the agents are fully developed and are deployed in the organisation, the predictive simulations that they could perform should be as automatic as possible (otherwise human intervention would make this anticipatory mechanism inefficient). This observation makes the need for better automatic alignment mechanisms very relevant.

Our future research will be directed towards a number of issues. First, mechanisms for triggering of the escape mode should be investigated, but also what the human will do after the escape is activated, i.e. how to train the agents. Other ways for alignment will also be investigated, like a priori alignment, which can be realised by a superior agent, or even by the agents themselves through a special kind of "pre-alignment interaction" - that would entail negotiation. Superior (software and human) agents can also intervene for alignment on-the-fly (a special kind of escape), and align the behaviour from one agent perspective, or impose a central solution from an global interaction perspective.

Second, for successful enactment of the alignment mechanisms, the way agents can choose an alignment policy has to be investigated. This raises a number of interesting questions. First of all, is an alignment policy a belief? Most likely it is not, as an alignment policy does not contain information about the environment. However, the information used for choosing a policy is based on beliefs about the environment, thus the decision mechanism is probably a belief. Another question is if agents can exchange alignment policies and how she chooses beliefs, and in this way learn from each other new ways of alignment. Such exchanges are regulated in agent societies by trust mechanisms, which means that an explicit representation of trust is needed. Finally, it is needed to figure out how agents can adapt their beliefs about the use of alignment policies.

5 Conclusion

As we have shown, it is possible to describe a policy for alignment that can be applied when the interaction beliefs of two or more interacting agents are not matching. We introduced an extension of Petri Nets to capture the interaction beliefs and also a mechanism to choose the appropriate policy that adapts the beliefs from one agent perspective. From the anticipatory systems perspective, this research can enable predictive agent model execution (agent-based simulation of organisational models) to be more reliable and necessitate less human intervention in terms of alignment.

We believe that interdisciplinary work between the BIS research and anticipatory agent research can yield lots of "cross-fertilisation" and raise the awareness that BIS enabled organisations are in fact anticipatory systems and also provide test beds for novel anticipatory agent ideas.

References

1. W.M.P. van der Aalst. Verification of Workflow Nets. In P. Azema and G. Balbo, editors, *Application and Theory of Petri Nets 1997*, volume 1248 of Lecture Notes in Computer Science, pages 407-426. Springer-Verlag, Berlin, 1997
2. W.M.P. van der Aalst, Interorganizational Workflows: An approach based on Message Sequence Charts and Petri Nets, *Systems analysis, modelling, simulation*, vol. 37(3), 335-381, 1999
3. W.M.P. van der Aalst and S. Jablonski, Dealing with workflow change: identification of issues and solutions, *Computer systems science and engineering*, vol 5, 267-276, 2000
4. P. Davidsson, E. Astor and B. Ekdahl, A Framework for Autonomous Agents Based on the Concept of Anticipatory Systems, *Proc. of Cybernetics and Systems 1994*, pp. 1427-1431, World Scientific, 1994
5. B. Ekdahl, Agents as Anticipatory Systems, World Multiconference on Systemics, *Proc. of Cybernetics and Informatics (SCI'2000) and the 6th International Conference on Information Systems Analysis and Synthesis (ISAS'2000)*, Orlando, USA, July 23-26, 2000
6. C.A. Ellis and K. Keddara, A Workflow Change is a Workflow, *Lecture notes in Computer Science*, vol 1806, 201-217, 2000
7. R. Hamadi and B. Benatallah, Recovery Nets: Towards Self-Adaptive Workflow Systems, *Lecture notes in computer science*, vol 3306, pp. 439-453, 2004
8. K. Jensen. An Introduction to the Theoretical Aspects of Coloured Petri Nets. In J.W. de Bakker and W.-P. de Roever, editors, *A Decade of Concurrency, Reflections and Perspectives*, volume 803 of Lecture Notes in Computer Science. Springer-Verlag, 1993
9. C.A. Petri. Kommunikation mit Automaten. PhD thesis, Institut fur instrumentelle Mathematik, Bonn, 1962
10. Gijs B. Roest, Nick B. Szirbik, Intervention and Escape Mode, *Proc. of AOSE workshop, AAMAS'06 Conference*, Hakodate, Japan, to be published in LNCS, http://tbk15.fwn.rug.nl/tal/downloadfile.php?id=31
11. R. Rosen, Anticipatory Systems. New York: Pergamon, 1985
12. K. Salimifard and M. Wright, Petri net-based modelling of workflow systems: An overview, *European journal of operational research*, vol 134(3), 664-678, 2001, Salimifardto appear, 2006
13. Cees de Snoo, Marco Stuit, Nick Szirbik, Interaction beliefs: a way to understand emergent organisational behaviour, to be published, http://tbk15.fwn.rug.nl/tal/downloadfile.php?id=54
14. http://tbk15.fwn.rug.nl/