INTERACTION BELIEFS
A Way to Understand Emergent Organizational Behaviour

Marco Stuit, Nick B. Szirbik and Cees de Snoo
Department of Business & ICT, Faculty of Mananagment & Organization, University of Groningen, Landleven 5
P.O. Box 800, 9700 AV Groningen, The Netherlands

Keywords: Agents, business process interactions, protocols, agent beliefs.

Abstract: Business processes consist of sets of individual and collaborative activities performed by agents (human of artificial) and the interactions between them. The agents have their own local beliefs and expectations about the behaviour(s) of the other agents. We represent these beliefs by using the 'interaction belief' concept. We show how a designer can reason about an interaction belief, how it can be modelled and how it is constructed for the purpose of simulation and agent development. Differences between workflow modelling and agent-oriented modelling are discussed. In order to illustrate the operation of the new concept, we present a business interaction example that shows how agents, equipped with interaction beliefs, can enact a business process in a non-centralised, emergent manner. Finally, we explore some interesting future research topics that have arisen due to the introduction of the interaction belief concept.

1 INTRODUCTION

This paper presents the concept of agent Interaction Belief (IB), as it is used to model and enact distributed business processes (BPs) in an (inter-) organizational environment. In order to define the IB concept, we use known concepts as *agent* (i.e. human agent, software agent), as it is used in modelling and system implementation, and the concept of *role* as it is used in organization theory. Some agent approaches propose the use of the interaction concept, for example, in MESSAGE (Eurescom 2000) and AORml (Wagner 2003) interaction diagrams are defined. The problem with these approaches is that the view is *centralistic* and agents have to obey an external description about how they should interact.

The IB is a concept and operational construct in our language named TALL (The Agent Lab Language, see Stuit & Szirbik 2006). This agent-oriented language is used for modelling BPs and for interactive simulation (gaming) in the AGE (agent growing environment) framework (Roest and Szirbik 2006). We use this framework for eliciting requirements for software agents that support BPs and to develop them in an iterative and incremental manner (we call this process "growing" the agents). Gaming sessions with stakeholders and experts from a target organization lead to the representation of their behaviour(s) in IBs. This development process, by taking local views, is helping the organization's members' to better understand the intricacies of their BPs, visualised as sets of running interactions (Stuit and Szirbik 2006). The IBs can be shown at different levels of abstraction, as process descriptions (based on Petri nets), but also as symbols attached to the agents that participate in the interactions.

This paper is organised as follows: section 2 compares workflows to our approach. Section 3 presents an example of an agent interaction and shows how this can be regulated by an external protocol. Section 4 explains the IB as a departure from the centralistic workflow view; section 5 explains the diagramming technique and how agents can use IBs to enact emergent BPs themselves. Section 6 touches the issues of learning, change, and the IB lifecycle. Section 7 presents a discussion. Finally, section 8 concludes the paper.

2 BUSINESS PROCESSES AND AGENTS

Classic BP modelling assumes that an omniscient observer – in the persona of the modeller – exists and that (s)he can understand and identify all the relevant aspects in the modelled domain. This usually leads to a superficial and inherently incomplete analysis result that is not able to reveal

Stuit M., B. Szirbik N. and de Snoo C. (2007).
INTERACTION BELIEFS - A Way to Understand Emergent Organizational Behaviour.
In *Proceedings of the Ninth International Conference on Enterprise Information Systems - SAIC*, pages 241-248
DOI: 10.5220/0002358102410248

the "implicit" and/or "hidden" knowledge of the stakeholders that are involved in the modelling exercise.

According to Yan et al. (2004) the central control used by classic Workflow Management Systems (WfMSs) is very difficult to maintain for BPs in distributed enterprises, large companies, and especially for inter-organizational workflows. Each independent company makes its own decision on how to do its "piece" of business. Several other authors (Adams et al. 2003; Klein et al. 2000; Reijers et al. 2003; Van der Aalst 2000b and 2003) have concluded that traditional WfMSs are unable to provide adequate support for exceptions or procedural deviations from the process model. Only ad-hoc workflow approaches can handle deviations from the process model, which allows them to support more than just rigidly structured BPs. This approach is closer to the agent-oriented view because it can handle processes that have not been completely defined in advance. Loosely coupled inter-organizational workflow processes (Van der Aalst 2000a) are a way to model complex distributed BPs in which each business partner has full control over its own local workflow process. These local processes are then related through an overall interaction structure. Still, these approaches present a global model perspective in which the modeller acts as a sort of 'overseer'. All indicate a priori the execution procedure for the interaction, as it is enforced on the participants and a lack of consistency between the behaviour of the agents is not allowed. Such a centralistic modelling view lacks the flexibility and changeability necessary to capture the ever-changing nature of processes in dynamic organizations (Stuit and Szirbik 2006).

In traditional workflow modelling, there is a distinction between the process definition phase and the process execution phase. When processes – named in this context "process instances" – are created dynamically, like in our AGE framework, this distinction becomes less explicit because the set of activities in the process instance and the sequence of activities are unknown beforehand. An agent can see its own local behaviour as a workflow representation, capturing the way it interacts with other agents, but from a local perspective. The main difference that the agent paradigm makes with workflows is that it allows for conflicting process views. Due to their control abilities (i.e. agents are considered autonomous decision makers) all agents have full control over their local part of the overall process. If the views of the agents that carry out the process are diverging too much, the agents will need to engage in a negotiation that leads to a solution for successful completion of the overall process. Our approach is an extension to workflows, but is opposed to a centralistic view, thus allowing change and adaptation within the organization (or network of organizations). The main shortcoming of a centralistic approach is that change can spread without knowing how far, because the concept of locality does not exist. Moreover, the agents will have more freedom in choosing their behaviour. In an agent approach, a central process definition would make no real sense anyway, because the central view has to be stored within every agent. This implies unnecessary redundancy, continuous global belief maintenance and an agent organization that has one single (process) belief; all these features being not at all characteristic for an agent-oriented approach.

However, workflow centralistic solutions have their advantages and a completely decentralised agent approach lacks a coordination mechanism, which could make such systems unstable and unreliable. Another problem is that, due to the lack of an explicit central definition, "optimization" of the BP is difficult to achieve. We do not advocate a "pure agent" approach – but we position our approach in the middle, between the workflow approach and the (pure) agent approach.

3 ENFORCING COHERENCE IN AGENT INTERACTION

Consider the typical inter-organizational process of negotiating the sale of a product between a seller and a buyer. If we describe the two organizations as agents, we can separately identify the buying and selling behaviour of the agents as process definitions. These definitions are only weakly prescriptive, the agents should always be able to change their local process definitions on the fly. Agents have always an epistemic (and auto-epistemic) dimension; they have beliefs about the other agents with whom they interact (and sometimes "beliefs about the beliefs" of the others). We can attach a supplementary description of how the agent believes that the other(s) will behave to the local process description. In this way, the agent has an overall process description of the whole interaction process. However, this contains *only* beliefs about the expected behaviour of the other agent(s), but there is no certainty that the real local process description of the other is matching the expectation of the first agent.

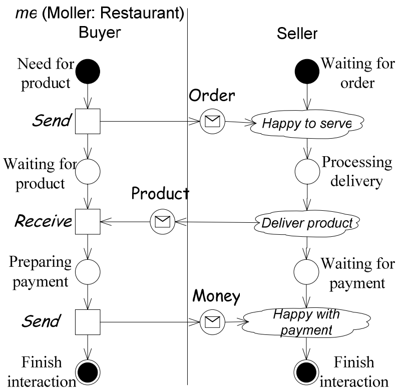

Figure 1: How the buyer sees the process.

We model first the buyer's overall process description (see fig. 1) by using an extension of Petri nets, named Behaviour Nets (Meyer and Szirbik 2006). Slightly different approaches to use (several types of) Petri-nets to model agents' behaviour can be found in Köhler et al. (2001) and Xu and Shatz (2001). Köhler et al. (2001) for instance use reference nets (so-called higher coloured Petri nets based on the "nets within nets"-paradigm) for graphical modelling of the behaviour of autonomous and adaptive agents including the exchange of messages with other agents. The process description for the "buying interaction" contains two swimlanes. One contains the intended activities to be performed by the buyer. The other swimlane describes what the buyer believes (and expects) the seller will do. In order to emphasise that the nature and order of the activities of the seller are uncertain, we depict these with clouds.

Swimlanes identify not only activity areas, but also the identity of the interaction participants. This concept is known in many modelling methodologies. We label the swimlane with a "role-name". In workflows, there is no local point of view. Because the diagram in figure 1 illustrates local behaviours and beliefs, it is necessary to show from whose point of view the diagram is made (i.e. who is responsible for this behaviour and belief). There are many potential choices to show this; for example, the first left-side swimlane is the one illustrating the "me" behaviour. We opted for adding supplementary information to the role name that identifies a swimlane, in the form of a predicate *me*, with the arguments *agent id* and *agent type*. Although the "cloud" notation already captures the nature of the swimlanes representing "others", this allows the analyst to see immediately who is responsible for this process description, and to see what type of agent it is.

Figure 1 shows that the buyer who is the *me* here first orders the product by sending a message to the seller. We have defined a small extension to the Petri net syntax, by introducing a "message place". The sender of a message is explicit from the incoming arc, and the receiver is explicit from the outgoing arc, but also from the position of the message place. The receiver will have the message place always positioned on its swimlane. Formally speaking, this is redundant also, but it enhances visibility for the analyst. Next, the buyer waits for the delivery of the product. If it receives it, it will inspect it and we assume in this (rather simplistic) model that it will accept it. After, it will send the money to the seller. The buyer believes that the seller should always send the product first and receive the money after.

Figure 2 shows how a seller who is the *me* here sees the process. It is waiting for an order and after the order is received, it will send an invoice to the buyer. It expects that the buyer will pay, and only then it will send the product. It is obvious that two "rigid" agents with these kinds of beliefs and behaviours will never succeed to complete the sales interaction. Fortunately, there are multiple solutions to this problem.

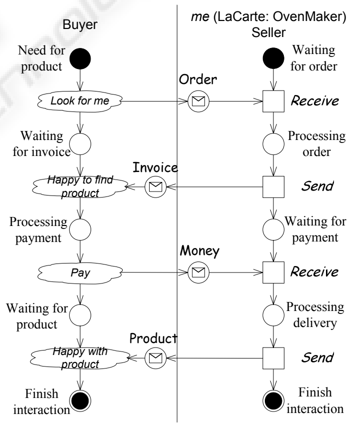

Figure 2: How the seller sees the process.

For example, one agent can unilaterally alter its behaviour when it notices that the behaviour of the other is different from its belief. The buyer may contact the seller and ask how the process is seen there. If it really wants the product, it can accept to pay first. In the same way, the seller can accept the payment after the product has been delivered. However, local change cannot insure that the overall process will end successfully. For example, if the seller accepts to send the product first, it is possible that the buyer will also change and have a decision point after receiving the product: (1) sending the money or (2) sending back the product due to a negative product inspection. This raises the necessity to have still a central point of view, from where the coherence of the overall process can be achieved. The typical solution for having a coherent interaction is to use an external mechanism that ensures the process will end correctly. The participants can accept a simple inter-organizational workflow description that exists a priori as a protocol (Lee 1999). A protocol-based solution for the buyer-seller interaction is given in figure 3. Here it is possible for the buyer to reject the product, and it is possible for the seller to receive the money before sending the product.

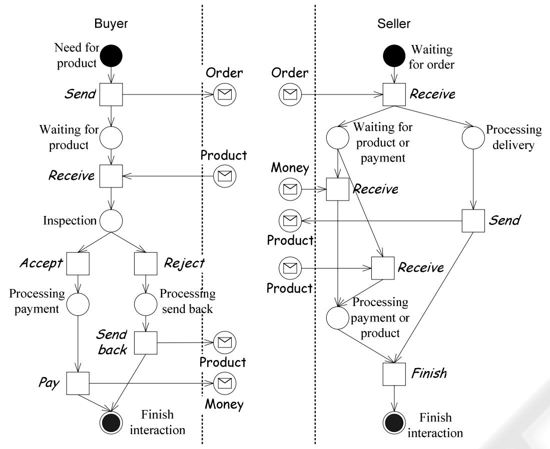

Figure 3: A protocol-based solution for the interaction.

However, in this situation the agents have to overrule their behaviours and beliefs and adopt "blindly" the given protocol. Still it is possible that in certain exceptional contexts, these protocols will not work (e.g., money is insufficient, the product is different, etc). The main objection to such an "imposed protocol" scheme is that it assumes that a third party providing protocols exists and can deliver protocols that are covering all exceptions for all the participants involved. In many situations, the "blind adoption" of the protocol can conflict with the way business is done by the participants. Following an uncustomary procedure means sometimes that the local BPs can be negatively affected (not always, "best practices" – on which protocols are usually based – can improve things). Protocols also tend to become obsolete fast, especially in dynamic business environments. Moreover, if the results of applying the protocol are highly negative, there is the issue of responsibility. Who is responsible, the participant using the protocol, or the third party who offers it? In practice, there is no answer to this question. Typically, if things turn out well, the participant will consider this normal and assume that the participant itself produced the positive effect. If things go wrong, the blame is on the imposed procedure and the third party.

4 FROM PROTOCOL-BASED TO LOCAL VIEW SOLUTIONS

We discuss here three solutions to avoid the use of protocols. First, participants can start an interaction without knowing if there is a chance that it will end successfully. Alignment of the behaviours of the actors happens 'on the fly'. Each time a problem appears, e.g. misunderstanding about messages, the actors start a negotiation-based interchange about their expectations and local views. Obviously, these "conversations" can take a lot of time, without any guarantee that a solution emerges. However, in completely new situations, this kind of interaction is widely used. Research to find a formal way to implement this kind of alignment – by using automatic resolution for software agents – delivered some initial results (Meyer and Szirbik 2006). Comparable with this approach of using alignment procedures to match different interpretations of BPs, Brockmans et al. (2006) present a semantic alignment approach (by means of ontology alignment) for Petri nets as opposed to existing only syntax-based alignment approaches.

The second solution looks for an a priori *belief exchange* and negotiation *before* the interaction starts. The participants try to formulate together new local behaviours that will be followed during the upcoming interaction. Especially in situations of long-term supply chain collaboration, organizations are using this type of alignment. Local views of performing the BPs, the internal culture and structure are communicated and discussed before a solution for the overall process is enacted. The drawbacks of this solution are that still a mechanism that checks the coherence of the overall process has to be used, and also negotiations can be very complex and time consuming – due to cultural differences for instance. For interactions that occur often and are performed in dynamic contexts, it is inefficient to start each time a complex alignment negotiation. However, in stable environments, this approach is appropriate.

For the third solution, we assume that within the environment of these two agents there is a third agent. This agent has the experience (behaviour) to solve their problem, in a way that will not adversely affect their behaviours (and subsequently their internal BPs). This solution uses a mediator that acts between the two participants, i.e. becomes part of the interaction. For instance, a bank can intervene between the buyer and the seller. In figure 4, we present how the bank can play a new role in the interaction. We assume that the bank "knows" about typical buyer-seller behavioural conflicts, due to its experience, and that it is willing to intervene, knowing also the identity and reputation of both the buyer and the seller. The interesting aspect is that only the bank "knows" what to do. We presume that the other two participants are not aware of a solution, although it can be applied straightforwardly.

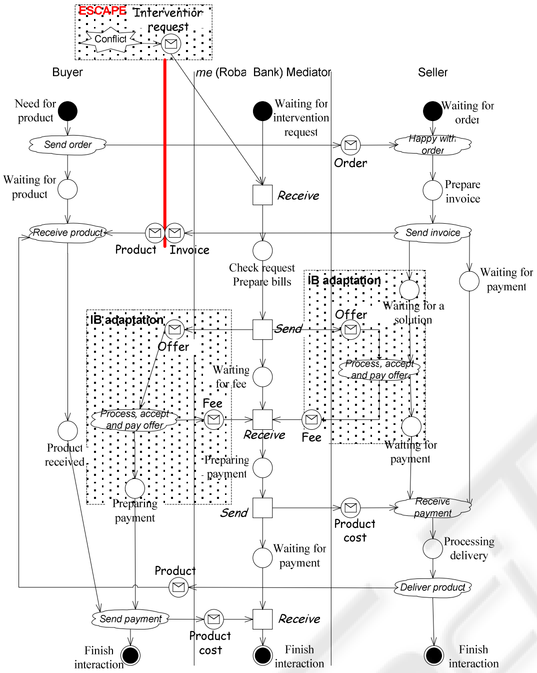

Figure 4: The Interaction Belief of the Mediator.

Our main point is that an interaction is just another emergent behaviour of various actors, *who do not have to know in advance exactly how the interaction will occur*. It may be that slight adaptations of the local behaviours should happen. Besides, it is possible that more participants than initially sought will be part of the interaction.

This is why we advocate for a different way to model the interaction. Because an interaction is just an emergent phenomenon from a set of local behaviours, we consider that the modelling should focus on those beliefs of the participants that refer to the interaction procedure. Of course, it is not trivial to grasp exactly what is a "belief about a specific interaction", either in the mind of an individual or the complex mesh of activities that result in the behaviours of an organization. However, the modeller can study via gaming (simulated interactions) the local behaviour of one of the participants (Roest and Szirbik 2006). (S)he may identify how this participant thinks about what it has to do, but also what the others are supposed to do, *according to the participant*. If most of these beliefs are collected from the participants in a specific (inter-organizational or inter-departmental) BP, one can easily reconstitute some process instance in a gaming-like way by playing the different roles that drive the BP. Due to the inherent inconsistencies in the collected views of the participants, the actor(s) playing the roles will have to adapt the local behaviours previously described. Although a human actor can do this extremely easy, it becomes a challenge to implement it within a simulation with software agents.

5 DIAGRAMMING INTERACTIONS

Because our main research push is towards modelling and simulation in agent-oriented environments, it is important to find a proper method to capture the belief about interactions that are leading to the local behaviours. For this, we use the IB construct. An IB has two or more swimlanes that identify the participants that are believed to be involved in that interaction, from the local viewpoint of one of the participants. As in figure 1 or 2, this swimlane will be specified by the *me(instance identity; type of instance)* predicate. On each swimlane, a Behaviour Net will describe the activities and the routing of the activities for each participant. There is a conceptual difference between the *me* swimlane and the other swimlanes. The former describes what the agent intends to do, thus we call this part of the IB an "intended behaviour" (abbreviated MB, from "My Behaviour"). The latter describes what the agent expects the others do, and we call this part the "expected behaviours" (EBs). For each believed participant, there will be one EB. Concisely:

1 IB = 1 MB + nEBs

As explained before the cloud symbol is used to depict that EBs are uncertain (like in fig. 1, 2, 4, or 6). A "cloud" indicates one (or more) activities performed by the "other" in a way that consistently mirrors the own local behaviour (MB). From the point of view of the agent that performs that swimlane in reality, the actual content of the cloud can be very complex, including multiple routings and/or other interactions that are transparent from the point of view of the *me* participant. In addition, such a cloud can involve the triggering of other interactions, a fact that is obscured in the belief of the *me* participant. The IB is a construct that appears to the modeller as one that is the ownership of a particular agent, because in reality an agent is the true owner of its own perceptions, convictions and beliefs about interactions in which it participates. A particular agent can own many identifiable IBs related to several interactions but some IBs can describe different beliefs related to the same kind of interaction,

We visualise the agents that interact by showing them and the IBs that are used to perform the interaction, like in figure 5. This is happening on a high level of abstraction, where the agent instances are the participants, and they appear like in AORml (Wagner 2003). The interaction (which is an intangible concept) is depicted as a double hollow arrow linked to the agents. On the lines that connect the agents to the interaction, we attach symbols (chevrons) that depict specific IBs that are owned and used by the agents in this specific interaction. These symbols are compact representations of IB diagrams. In the AGE simulation environment, where we visualise running interactions, one can for example double click on the chevron *b1*and obtain in another window the whole "swimlaned" IB diagram of figure 1.

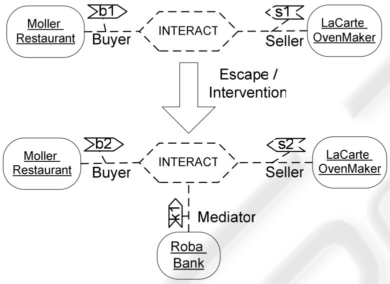

Figure 5: High-level view of the buyer and seller.

The agents "Moller" and "LaCarte" exist in an environment. If they enact a buyer-seller interaction by using the IBs of figures 1 and 2, the process will fail. In this case, either the buyer or the seller will detect the failure and go into escape mode. Next, the environment will give a solution. We depict the failure as a conflict in figure 4. The bank is triggered by the environment that knows the bank can provide a solution. Supposing that the conflict (the buyer receives an invoice instead of a product) is detected by the buyer, the bank will be informed and the interaction is re-started in a different way. The bank is sending its own IB to the buyer and seller – these messages are represented as part of the banks' behaviour. They can immediately see that the bank's EBs (on their part) are almost similar to their MBs, and only a minor adage to these swimlanes is necessary. The IBs of the buyer and seller are changed dynamically. By "accepting" (sending the messages depicted in fig. 4), they also show to the bank that their behaviour has been adapted to the new situation. We call the use of the environment in such a way "escape/intervention" (Roest and Szirbik 2006). The buyer agent invokes an external "authority" (in real life it can be a commercial consultancy, in our case it is a simulation environment) that is able to find a solution based on the behaviour of another agent. That means that this new agent also has to make known to the environment that it is able to intervene. In reality, escape situations are difficult to model and trigger. The typical escape situation is when the agent does not know what to do in an interaction – implemented as "time escalation triggers" in many systems, e.g. workflows. In these situations, human intervention is needed, and automation is difficult. ERP systems implement these by various exception handling procedures. Interventions in business interactions are usually highly regulated (by the rules and laws of the environment), but the way the problem is solved is ad-hoc and context dependent and most of the time needs human intelligence.

6 NEW BELIEFS AND THE IB LIFECYCLE

Agents, by adapting their behaviour during interactions, are implicitly learning new behaviours. These new IBs can be logged and used in other interactions. For example, the buyer agent, who has "been exposed" once to the banks' behaviour, can log the past interaction as an IB. Of course, this new IB (see fig. 6) will have the *me* swimlane on the buyer side of the diagram, and the rest will be "clouded". In this new IB of the buyer, new activities and places should be inserted in order to self-trigger the mediator involvement in the "conflict situation". In figure 6, these new components appear in the grey rectangle. If this interaction is started and another seller is involved (that does not know the possibility of the bank-mediated scenario), and the invoice will be sent by the seller, the conflict will be detected directly by the buyer. The buyer will use the "learned" IB and involve the bank. This time, the bank will not intervene because the environment triggered it to do so, but because another agent asked directly for its involvement. This IB still considers that it is possible that the seller will be "friendly" and send the product first. In this case, the buyer will dynamically change the running IB (as in fig. 6 – see the double-lined activities) with the one that is depicted in figure 1. In this "learned" scenario, the escape/intervention mechanism is not used, and the final overall process, considering that the seller will adapt like in figure 4 or 6, can be seen as an emergent phenomenon, based on the IBs of the agents.

An IB can reside within the belief base of an agent and has a life cycle. First, it can be built in an analytical way and made explicit by a modeller. Second, it can be in current use, either as an intention or a representation of the current state of the agent during a running interaction. Finally, it can be an explicit log of a successful past interaction. The log can be used again when the agent considers this worthwhile.

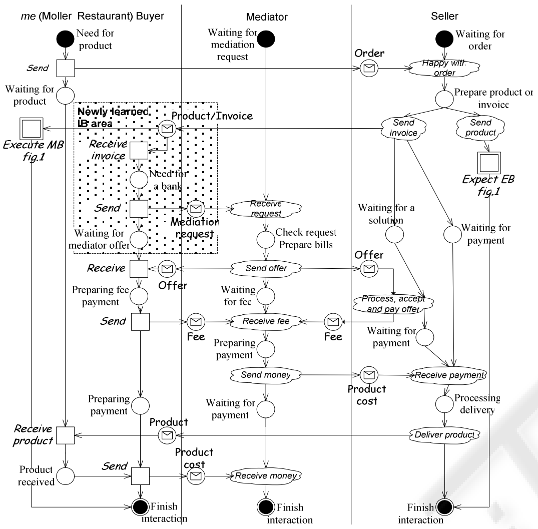

Figure 6: The newly learned IB of the buyer.

In the multi-agent simulation environment, the experimenter and/or the players should be able to visualise the IBs in their various points in the lifecycle and manipulate them when necessary. Diagrams in the editing and simulation windows of the AGE environment look like the ones presented in this paper, but supplementary information should be given about the lifecycle point. When an interaction is underway, it should be possible to attain a coherent view of the overall process, that is, a process diagram that cumulates the distributed IBs in one (central) view. Of course, the activities on the swimlanes have to be aligned automatically or manually, in order to obtain a coherent process description that implements the interaction.

7 DISCUSSION AND FUTURE WORK

An approach based on IBs raises a series of questions related to a multi-agent simulation environment. Such a tool should allow changes in the IBs of the agents. These changes can either be operated manually, like the change in the IB of the buyer, or can be automatically implemented via machine learning mechanisms. The main idea is that the behaviour of the players is captured – manually or automatically – in the form of the IBs during the "business process" interactive gaming. In this way, the simulation itself will become more and more automatic, because the agents enrich their behaviour and will be able to automatically perform more interactions. However, the human will remain always an intervention factor when the agents are triggered into escape mode. More about this behaviour capturing is described in Roest and Szirbik (2006).

A future research issue is when a human agent is leaving an organization and enters another one. The software agents that supported it in various interactions remain in the organization and will have explicit IBs that reflect the past behaviour of the leaving human. A new human agent that will play the same roles will be able to learn from these IBs and even use them (in an adapted form). When the behaviours are routine and stable it is possible that there is no need for a new human at all. This always happened when information systems made clerical staff redundant. The human who left for another organization carries explicit IBs that can be used again by the human or by software agents, if the supporting infrastructure exists. We want to investigate the link between this migration of behaviour and the innovation processes in organizations. In real life situations, innovation occurs when a new manager uses positive (but also negative) past experience in another organization, in order to improve BPs there.

One of the crucial features of the agents that use will be their ability to learn. They have to be able to select the appropriate IB from their belief base. Specific methods from artificial intelligence can be employed here. Process mining (Van der Aalst et al. 2003) will be used to discover interaction patterns from the IB logs. Alignment policies also have to be built in machine learning fashion. Organizations change, new roles and participants appear. We want to investigate the impact of these changes on the agents that use IBs with fixed role names attached to the swimlanes. In the past such role names tended to be very stable but nowadays change occurs more frequently. A way to decouple the roles from the organizational hierarchy is to link them to "embodied interactions". This means that in figure 5, the interaction symbols become concrete entities, that capture the nature of the BP of the organization, and the roles will be placeholders for the agents. More about this line of research can be found in Stuit and Szirbik (2006). All of these issues are immediate on our list of future research topics. Finally, we want to compare our methodology with others, both in BP analysis/(re)design and in agent-oriented software engineering frameworks. Candidates are MESSAGE (Eurescom 2000), which has an explicit interaction specification, Tropos (Bresciani et al. 2004), and Prometheus (Padgham and Winikoff 2003).

8 CONCLUSION

We strongly believe that it is more effective to capture human behaviours by focusing on interactions and build bottom-up an emergent process, which is a result of much autonomous behaviour. Real organizations and humans work in this way. As pointed out by Ekdahl (2000), any agent approach is justified when the architecture of the software agent system (for both organizational simulation and BP support) is inspired from a social situation. Human and organizational behaviour are the real drivers that lead to the emergence of the BPs within organizations and networks of businesses. We consider that any (multi-) agent approach should find some way to capture the local behaviour of the interacting agents. Our idea to represent this behaviour as Interaction Beliefs via a Petri net extension is new and promising. It is inspired from inter-organizational workflows and agent architectures, and can apply the principle within a single organization, as well as within networks of organizations. This also allows us to model BPs that are otherwise very difficult to represent via the "classic" centralistic methods.

REFERENCES

Adams, M., Edmond, D., & Ter Hofstede, A.H.M., 2003. The application of Activity Theory to Dynamic Workflow Adaptation Issues. *Proceedings PACIS-03, 1836-1852.*

Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., and Mylopoulos, J., 2004. An Agent-Oriented Software Development Methodology. *Autonomous Agents and Multi-Agent Systems,* 8 (3), 203-236.

Brockmans, S., Ehrig, M., Koschmider, A., Oberweis, A., and Studer, R., 2006. *Proceedings ICEIS 2006, 191-196.*

Eurescom, 2000. *Message: Methodology for Engineering Systems of Software Agents, Deliverable 1, Initial Methodology*. Project P907-GI.

Ekdahl, B., 2000. Agents as Anticipatory Systems. *Proceedings SCI 2000 and ISAS 2000, 133–137.*

Klein, M., Dellarocas, C., & Bernstein, A., 2000. Introduction to the special issue on adaptive workflow systems. *Computer Supported Cooperative Work,* 9 (3-4), 265-267.

Köhler, M., Moldt, D., & Rölke, H., 2001. Modelling the Structure and Behaviour of Petri Net Agents. *LNCS, 2075, 224-241.*

Lee, R.M., 1999. Distributed Electronic Trade Scenarios: Representation, Design, and Prototyping. *Electronic Commerce,* 3, 105-136.

Meyer, G.G., & Szirbik, N.B., 2006. Behaviour Alignment as a Mechanism for Anticipatory Agent Interaction. *Proceedings ABiALS 2006.* Available from: http://www.rug.nl/staff/g.g.meyer/p4.pdf?as=pdf

Padgham, L., & Winikoff, M., 2003. Prometheus: A Methodology for Developing Intelligent Agents. *LNCS, 2585, 174-185.*

Reijers, H.A., Rigter, J.H.M., & Van der Aalst, W.M.P., 2003. The Case Handling Case. *International Journal of Cooperative Information Systems*, 12 (3), 365-391.

Roest, G.B., & Szirbik, N.B., 2006. Intervention and Escape Mode: Involving Stake-holders in the Agent Development Process. *Proceedings AOSE-06, 109-120.* Available from: http://www.rug.nl/staff/g.b.roest/GijsBRoest&NickBSzirbik_Escape_Intervention.pdf?as=pdf

Stuit, M., & Szirbik, N.B., 2006. Embodied Interactions: a Way to Model and Execute Complex and Changing Business Processes with Agents. *Proceedings ESAW06, 169-184.* Available from: http://www.rug.nl/staff/m.stuit/p3.pdf?as=pdf

Van der Aalst, W.M.P, 2000a. Loosely coupled interorganizational workflows: modeling and analyzing workflows crossing organizational boundaries. *Information & Management,* 37 (2), 67-75.

Van der Aalst, W.M.P., 2000b. *Exterminating the Dynamic Change Bug: A Concrete Approach to Support Change.* Beta Working paper Series, WP 51, Eindhoven University of Technology.

Van der Aalst, W.M.P., 2003. Inheritance of Business Processes: A Journey Visiting Four Notorious Problems. *LNCS, 2472, 383-408.*

Van der Aalst, W.M.P., Ter Hofstede, A.H.M., and Weske, M., 2003b. Business Process Management: A Survey. *LNCS, 2678, 1-12.*

Wagner, G., 2003. The Agent-Object-Relationship metamodel: towards a unified view of state and behavior. *Information Systems,* 28 (5), 475-504.

Workflow Management Coalition (WfMC), 1999. *WFMC: Terminology & Glossary*. WFMC, Document Number WFMC-TC-1011.

Xu, H., & Shatz, S.M., 2001. An Agent-Based Petri Net Model with Application to Seller/Buyer Design in Electronic Commerce. *Proceedings ISADS-01, 11-18.*

Yan, Y., Maamar, Z., & Shen, W., 2001. Integration of Workflow and Agent Technology for Business Process Management. *Proceedings CSCWD-01, 420-426.*