

C-IPS: Specifying Decision Interdependencies in Negotiations

Kay Schröter and Diemo Urbig

Humboldt-Universität zu Berlin
Department of Computer Science
{kschroet,urbig}@informatik.hu-berlin.de

Abstract. Negotiation is an important mechanism of coordination in multiagent systems. Contrary to early conceptualizations of negotiating agents, we believe that decisions regarding the negotiation issue and the negotiation partner are equally important as the selection of negotiation steps. Our C-IPS approach considers these three aspects as separate decision processes. It requires an explicit specification of interdependencies between them. In this article we address the task of specifying the dynamic interdependencies by means of IPS dynamics. Thereby we introduce a new level of modeling negotiating agents that is above negotiation mechanism and protocol design. IPS dynamics are presented using state charts. We define some generally required states, predicates and actions. We illustrate the dynamics by a simple example. The example is first specified for an idealized scenario and is then extended to a more realistic model that captures some features of open multiagent systems. The well-structured reasoning process for negotiating agents enables more comprehensive and hence more flexible architectures. The explicit modeling of all involved decisions and dependencies eases the understanding, evaluation, and comparison of different approaches to negotiating agents.

1 Introduction

Consider having a paper presentation at a scientific conference late in the evening close to the end of the conference. Perhaps you are in conflict with several participants who leave the conference earlier, but you would like to talk to as many participants as possible. You might try to "negotiate" with them about staying until your presentation. Take care to recognize the conflict quickly; else they might already have been gone. You can also choose another negotiation issue with other partners to solve the conflict: You may negotiate with the local organizer or perhaps the session chair to assign you another time slot. Perhaps you are already in contact with the local organizer and you can take the chance to adapt the issue of conversation; hence, you do not have to set up another interaction. Again, it is important that you recognize the conflict on time.

If not the same, but similar situations associated with perceived conflicts may occur in complex multiagent systems. In high-level multiagent systems, which you can find in e-business environments as for instance Internet purchasing or supply chain management, negotiations are an often applied mechanism

to resolve such conflicts [HS99,San99]. Also in other domains negotiation is a frequently used metaphor for specific conflict resolution mechanisms. That is why analyzing and modeling negotiating agents has become an important area of research in distributed artificial intelligence and information systems engineering.

An influential conceptualization of analyzing and modeling negotiating agents distinguishes between three most important concepts: the negotiation object, the negotiation protocol, and the agent's internal reasoning process [JFL⁺00]. Although several authors have explored questions like which issue should be negotiated (e.g. [FWJ03]) and how are negotiation partners selected (e.g. [KL02]), these aspects have not been considered as generally important concepts in agent negotiation. A comment at an early presentation of the C-IPS framework revealed that sometimes scientists assume that the conflict itself unambiguously defines the partners that are needed to solve the conflict. But as can be seen by our introductory example, this is not a general law. The example shows that recognition of conflicts and selection of negotiation issues and partners are important processes when dealing with negotiating agents. They become especially important for complex open multiagent systems. Therefore, when analyzing different solutions to negotiating agents the answers to questions about issue, partner, and step selection processes should be easily accessible. But the already mentioned conceptualization does not address decisions related to a negotiation partner nor does it provide an adequate structure for the agent's reasoning process. Thus, also the (dynamic) dependencies between different decisions are not explicitly specified.

The C-IPS approach that we take up in this article is a promising way to structure negotiating agents [UMS03,US04]. It distinguishes between external constraints (C) the agent has to follow and an internal reasoning process (IPS) that describes the agent's decisions. The latter deals with decisions regarding the selection of negotiation issues (I), partners (P), and particular negotiation steps (S). The interdependencies between these three decisions can be described at a new separate level, which differs from dealing with negotiation protocols or defining negotiation objects. Based on a simple example we illustrate the specification of such interdependencies. The C-IPS approach does not only provide a more sophisticated, comprehensive, and explicit perspective to negotiating agents, but also it provides a better base to compare different designs of negotiating agents.

Following this introduction, the second section provides a brief summary of the C-IPS approach. For modeling IPS dynamics the third section starts with an introduction of some general states and predicates for these states. Subsequently we apply the general ideas to a simple example: the modeling and implementation of agents in the INKA project¹. In the fourth section we specify negotiating agents in a very idealized world, while the fifth section continues with extending it to capture more realistic phenomena in open multiagent system, i.e. drop out of partners or problem solution as side effects of other actions.

¹ The *INKA project* is funded by Deutsche Forschungsgemeinschaft (DFG) under grant 788/4-2 and 788/4-3 within the priority program *Socionics*. This priority program supports research projects that combine sociology and computer science.

2 The C-IPS Approach

A negotiation is a process by which a joint decision is reached by two or more agents, each trying to reach an individual goal or objective [HS99].

Consider an agent designer modeling a negotiating agent that should be able to solve a particular conflict. The agent might be enabled to decide everything that is necessary to solve the conflict in any way that is possible. Due to complexity reduction as well as due to the application domain this freedom is frequently restricted. Thus the C-IPS approach separates external constraints from the agent's internal reasoning process. The distinction between protocols and strategies (see [HS99,RZ94]) mirrors the same logic but claims less generality. The two driving forces of external constraints, i.e. application domain and agent designer, appear in [SW03] as exogenous and endogenous criteria.

The C-IPS approach considers decisions regarding the selection of negotiation issues, the selection of negotiation partners, and the selection of negotiation steps as separate although mutually dependent aspects of negotiating agents [UMS03,US04]. Hence, the external constraints as well as the agent's internal reasoning process contain these aspects as separate parts. The C-IPS approach encourages and requires an explicit specification of the interdependencies between these three parts. This article provides a first attempt to do this in a standardized manner. Before we specify the dynamics of our example, we briefly introduce the different parts of C-IPS. We relate them to the concepts of negotiation space and protocols and give an intuition why we believe the C-IPS approach to be more general while at the same time more specific.

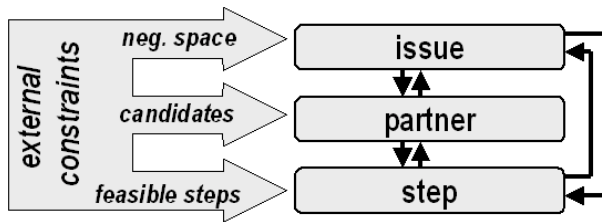


Fig. 1. C-IPS approach.

External constraints define the joint decisions agents are able to negotiate, i.e. the *negotiation space* N . If the external constraints for the negotiation issue are not too strong, then negotiating agents usually consider only subspaces in one ongoing negotiation, i.e. *negotiation issues*. For selecting a negotiation issue Huhns and Stephens suggest that agents communicate their position to identify conflicts, which then represent the issue of negotiations [HS99]. For flexible agents with large negotiation spaces a continuous broadcasting or a public knowledge of goals and objectives might be inefficient or even strategically inappropriate. Additionally, it can happen that the resulting space of possible agreements is too large, such that it should be reduced more. The reduction

of large negotiation spaces, i.e. selection of issues, is a very important and not trivial aspect of an agents' reasoning process, see for instance [FWJ03]. This reduction can even change (mostly shrink) the set of negotiation partners, especially if different partners are responsible for different issues. The introductory example about the conference gives an intuition for this problem. Keep also in mind that the negotiation space does not need to be fully explored. Because negotiations are not necessarily only about making concessions but can also be about exploring the negotiation space to find currently individually unknown possible joint decisions [HS99].

An appropriate balance between external constraints and internal decision processes is also vital for the choice of *negotiation partners* from the set of *candidates* C . If C is restricted to agents that suffer from a conflict, then superficial solutions might be lost, i.e. not every conflict unambiguously determines the appropriate negotiation partners. Again this is exemplified by our example. It is open whether a negotiation with conference participants is more promising than a negotiation with the local chair. Sometimes it might even be more appropriate to select negotiation partners before fixing a particular issue; for instance, a provider of several products with thousands of potential buyers might select the negotiation partner before fixing the product to sell.

Beside issues and candidates, agents need *interactions* to reach a conflict resolution. Frequently these interactions are communicative acts or sequences of them. From the set of *feasible steps* F given by a protocol the agent selects the actually intended *negotiation step*. The step component may also select sequences of steps or even strategies; this depends on the specific implementation. Protocols may also restrict the set of candidates. However, one can imagine that agents think about the appropriate protocol or about the protocol a partner actually applies. The protocol and the selection of steps according to the protocol are up to now the best analyzed aspects for negotiating agents; the other aspects and especially their interdependencies are less well analyzed.

Although the C-IPS approach encourages a more precise specification of interdependencies between the selection of issues, partners, and negotiation steps, we emphasize that it does not require a specific implementation of selection processes nor of their interdependencies. The decision can follow a sequential approach or agents may be allowed to change the issue in an ongoing negotiation or to choose the partner before fixing a negotiation issue or even the other way around. The decisions themselves may be implemented following a complex BDI approach or other approaches, but this is not subject to this article. Thus C-IPS provides a level of analysis and modeling that is above designing negotiation protocols and strategies.

The C-IPS approach has been applied within the INKA project. The project aims at the development of socially intelligent agents that negotiate the exchange of shifts in a hospital. The project's objective is not to contribute to theory on scheduling but analyzes the human behavior related to the application of socially intelligent agents [MUSG03]. INKA agents can automatically initiate and perform negotiations on behalf of their users, but the users can also make

all decisions themselves. During the analysis and modeling phase the C-IPS approach has successfully supported the communication between sociologists and computer scientists. Additionally, it has provided a basis for locating aspects of negotiating agents that are affected by specific sociological concepts [UMS03].

3 Specifying IPS Dynamics with State Charts

After introducing the C-IPS framework, we now come to the main contribution of this paper: the specification of interdependencies between the issue, partner and step selection processes. We will do so by defining states and transitions between these states. State charts will be used to visualize the specification. Thereby we build upon and extend the concepts developed in [US04]. Before we come to our example, we will provide some generally required states, predicates and actions.

The most prominent kind of state in our approach is an *IPS state* accumulating the decisions made by the different components. We define an *IPS state* as a three-dimensional vector $(I, P, S) \in (N \cup \perp) \times (C \cup \perp) \times (F \cup \perp)$. This state can be interpreted as the intention of the agent to act in a particular way regarding a given issue with a given partner. The three slots may have specific values or may be undefined; the latter is indicated by a \perp . If a specific value is set, it must be in accordance with the relevant external constraints given the other values in the *IPS state*. The different slots of an *IPS state* can be set via the actions² $I = SEL_I$, $P = SEL_P$, and $S = SEL_S$, respectively. These actions model the different selection processes. If no decision is possible then the actions result in undefined values \perp .

For *IPS states* we define the predicate $DONE(I, P, S)$ that holds if the current *IPS state* represents an intention that has been already performed. The agent may realize that particular *IPS states* are – at least temporarily – impossible; it is not useful to consider these states for decisions in the near future³. Hence, we need a mechanism to check whether a set of *IPS states* is impossible and we need a mechanism to mark a set of *IPS states* as impossible. The predicate $IMP(I, P, S)$ holds if a state is currently impossible. The action $SET_IMP(I, P, S, t)$ marks the state (I, P, S) impossible; this impossibility is reconsidered after a time period specified by t . An asterisk instead of a specific I , P or S value stands for an arbitrary element. Thus a whole set of states is checked or marked; for instance, $SET_IMP(I, *, *, 20)$ sets issue I impossible with any partner and step for the next 20 time units.

4 Example: IPS Dynamics for a Sequential Approach

So far we provided generally required states, actions and conditions. We will now illustrate the specification of IPS dynamics for a sequential approach. This

² These are actions in the terms of state charts. They are not agent interactions.

³ Imagine the agent does not find negotiation partners for a specific issue or all possible negotiations on that issue failed. Then all states comprising this issue are temporarily impossible.

approach has been used when designing our INKA system. First we will explain the effects of some external constraints we have set when designing our example. These constraints are not domain dependent and are not implied by the C-IPS framework. Rather the C-IPS framework allows specifying the negotiation process for many different constraints. Second we will give the state chart for this approach in an idealized environment. Third we extend the state chart to deal with the challenges of an open multiagent system.

4.1 External Constraints Influencing the Dynamics

As a first constraint, we follow a sequential approach in taking the three decisions regarding the negotiation. We start with selecting an issue, then we chose an appropriate partner for that issue and finally we decide on the next step in the negotiation on that issue and with that partner. This implies that all transitions between states can only change one slot of an *IPS state*. The sequential approach reduces the number and the complexity of interdependencies, which results in fewer transitions and less complex conditions for the transitions.

Second, our example requires that an agent is only engaged in one negotiation at the same time. We define an agent as engaged in a negotiation if it has selected a step S . For specifying the interdependencies we need to distinguish between *IPS states* that allow joining a negotiation and *IPS states* that do not so. That is why we introduce the two states *busy* and *-busy*. The state *busy* comprises all *IPS states* where the step S is specified and the state *-busy* comprises all other *IPS states*. If the agent is in state *busy* it will not participate in any further negotiation. As the transitions between *busy* and *-busy* are closely related to transitions between *IPS states* they are not explained here, but in the following subsections.

Third, in our system the information distribution is asymmetric. The agents do not know about each other's conflicts. This requires that our agents are able to ask other agents to help them by negotiating the conflict. We term an agent that asks for a negotiation *initiator* and an agent that accepts the request *responder*. An agent must be able to request a negotiation (via step S_r) and to agree to a negotiation (via step S_a). Contrary to the initiator it is not reasonable to assume that the responder can reuse slots of the *IPS state* after the negotiation has been finished⁴. Hence the *former responder* removes all slots of the *IPS state*

Fourth, we require our agents to accept incoming requests for negotiation (*RCV_REQ*) except they are already handling another request. In that case the request is rejected because of becoming or being busy (*SND_BUSY*). We term an agent that is not yet busy and has not received a request for negotiation a *potential initiator*. An agent that is not yet busy but did receive a request is a *potential responder*.

For specifying the dynamics we define five additional states: *pot_ini*, *ini*, *pot_res*, *res*, and *frm_res*, which represent above mentioned five different roles

⁴ While the initiator may want to try a different partner on the same issue after a failed negotiation, the responder wants to solve its own conflicts.

an agent can take: *potential initiator*, *initiator*, *potential responder*, *responder*, and *former responder*. We handle these states in a separate role state machine (see Figure 2). This reduces the complexity of the conditions in the negotiation state chart, which specifies the IPS dynamics.

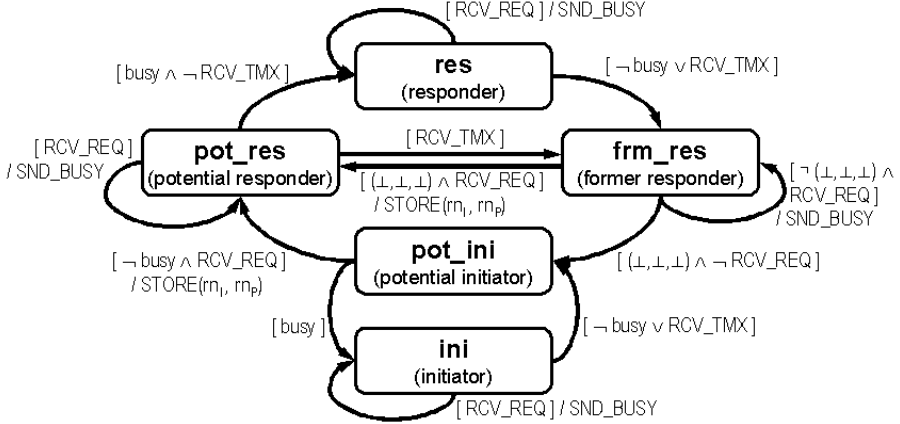


Fig. 2. The Role state chart.

Initially an agent is a potential initiator (state *pot_ini*). If it becomes busy without being requested before, then it is an initiator (state *ini*). After it is not busy anymore, e.g. the initiated negotiation has been finished, it switches back to the state *pot_ini*. If an agent in state *pot_ini* receives a request for negotiation while it is not yet busy, then it stores the information from the request ($STORE(rn_I, rn_P)$) and becomes a potential responder (state *pot_res*). Once it got busy by doing the agree negotiation step (S_a) it is in state *res*. After the accepted negotiation has been finished, the responder is not busy anymore and becomes a former responder (state *frm_res*). Finally it reaches the empty *IPS state* (\perp, \perp, \perp). If there is a new request the agent stores the corresponding information and directly changes to state *pot_res*, otherwise it changes back to the initial state *pot_ini*.

Only in state *pot_ini* the agent starts handling a request for negotiation. In all other states requests are rejected. As the reception of a request in state *pot_ini* results in a state transition, it is ensured that always only one request is handled.

The condition RCV_TMX represents the reception of a timeout message from the negotiation partner. In open multiagent systems a timeout may occur because a previous message was lost, the delivery took too long, or the partner is not willing to wait any longer. A timeout may only occur in the states that relate to an active negotiation. In the states *pot_res* and *res* a RCV_TMX results in a direct change to *frm_res*. In state *ini* it causes a transition to state *pot_ini*. The timeout will become relevant only in subsection 4.3. We introduced it here to give a complete description of the role state chart.

4.2 Basic IPS Dynamics

Now we will describe the IPS dynamics of the example, i.e. a sequential approach in an idealized scenario (see Figure 3). Here we exclude any external events influencing the *IPS states* that may occur in an open multiagent system. Transitions between IPS states only happen due to negotiation related events. This restriction allows us to focus on the main path of dependencies between the negotiation decisions. The presented model is a further developed version of the one presented in [US04].

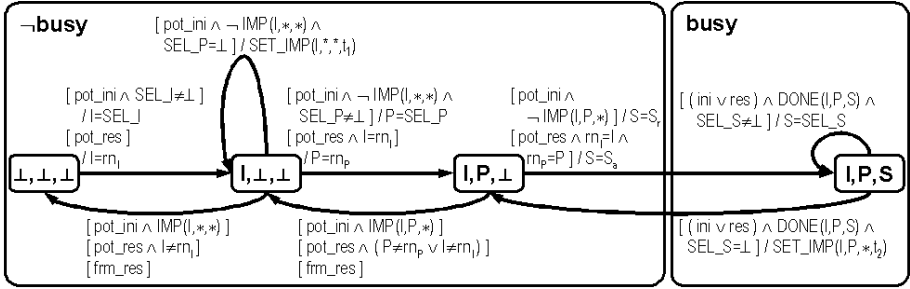


Fig. 3. Basic dynamics for the sequential approach in an idealized scenario.

If an agent is in state (\perp, \perp, \perp) there are two ways to change the issue. First, if the agent is a *potential initiator*, then the agent tries to select an issue by applying the corresponding selection function $I = SEL_I$. Second, if the agent is a *potential responder*, then it sets the issue according to the information stored, i.e. $I = rn_I$. In all other cases the agent remains in state (\perp, \perp, \perp) . If the *potential initiator* has already selected an issue but not a partner or a step, i.e. (I, \perp, \perp) , and it recognizes all states with this issue as being impossible, i.e. $IMP(I, *, *)$, then the issue is removed. The issue is also removed if the agent is a *potential responder* but the wrong issue is set⁵. Finally a *former responder* removes the issue to reach the empty *IPS state*.

In state (I, \perp, \perp) there are two ways to set a partner. If an agent is a *potential initiator*, and not all states with issue I are impossible then the agent tries to select an appropriate partner. If it cannot select a partner, then the partner component recognizes states $(I, *, *)$ as impossible $SET_IMP(I, *, *, t_1)$, otherwise the partner is set accordingly $P = SEL_P$. If the agent is a *potential responder* and the issue has already been set accordingly, then the partner is also set, i.e. $P = rn_P$. If a *potential initiator* has a chosen issue and partner but the state (I, P, \perp) is recognized to be impossible, then the partner is removed. It is also removed in case of being a *potential responder* but not having set the request-

⁵ This happens if the request for a negotiation arrives while not all components of the agent have finished the decision process.

ing agent as partner or the requested issue as issue. Finally a *former responder* removes the partner to come closer to the empty *IPS state*.

If state (I, P, \perp) is not impossible, after a *potential initiator* has chosen an issue and a partner it requests the partner to negotiate the selected issue; in fact, it sets a request for negotiation as the next negotiation step, i.e. $S = S_r$. If an agent is a *potential responder* and the issue and partner are set accordingly, then a requested agent agrees to negotiate the requested issue with the requesting agent, i.e. $S = S_a$. By following one of these two transitions the agent becomes *busy*. In state (I, P, S) , where issue, partner and step have been selected, an initiator or a responder can change the step $S = SEL_S$ after it has realized its intention, i.e. $DONE(I, P, S)$. If no new step can be found because a final step has been done, then an initiator or a responder recognizes a further negotiation with the partner on that issue as impossible $SET_IMP(I, P, *, t_2)$ and changes to state (I, P, \perp) . By doing that it becomes *¬busy*.

4.3 Extended IPS Dynamics

For this subsection on IPS dynamics we drop the restrictions made for the previous subsection: external events may now influence the IPS states. This is a realistic assumption for a complex distributed open multiagent system. It may happen that the negotiation partner or the negotiation issue disappears for some reasons⁶. For instance, the related conflict is solved as a side effect of some other action within the multiagent system. Because in such cases further negotiating this issue is useless, the agents should handle such situations in a flexible manner. The modeling of interdependencies resulting from such events requires a way to cancel a negotiation and a way to check whether a certain issue or step is still available. The first is a prerequisite for the negotiation protocol that has to enable a cancel of negotiations because of having no appropriate negotiation options S_c . The second is realized using a predicate $AV(X)$, where X is an issue or a partner. The predicate holds if the given argument is still available. The extended dynamics result from the basic dynamics only by adding some states and transitions. All states and transitions mentioned before are still included. In an open distributed multiagent system it may also happen that the potential or current negotiation partner may not want to wait any longer and sends a timeout. This has already been discussed in a previous subsection and is handled by the role state chart.

If a *potential initiator* or an *initiator* recognizes that the chosen issue is not available anymore $\neg AV(I)$, then it drops the issue. This may happen in states (I, \perp, \perp) , (I, P, \perp) , and (I, P, S) . The resulting transition for the first case does already exist; we just have to add another condition for it ($\neg AV(I)$). The second case is without any problems. Because there is no ongoing negotiation, the issue component can easily remove the issue; later, also the partner will be removed. In the third case there is an ongoing negotiation. Here the agent tells the partner that it cancels the negotiation because the continuation of the negotiation makes

⁶ A negotiation issue disappears when there is no possible agreement.

no sense, i.e. there are no alternatives to negotiate on anymore (S_c). After the cancel message has been sent, the partner component also removes the partner.

In an open multiagent system it may happen that an agent exits the system; hence it is not available anymore. In such cases, where $\neg AV(P)$ holds, the partner is removed from the IPS vector. This may happen in states (I, P, \perp) , (I, P, S) , (\perp, P, \perp) , and (\perp, P, S) . The transition for the first case is already covered by the basic dynamics. Again, only another condition has to be added. For the second case, if the partner component realizes a loss of partner but issue as well as step is defined, then the partner and afterwards the step are removed. The last two cases result from losing the issue before the partner. In these cases the partner can be dropped. While the agent is *not busy* only a *potential initiator* may lose the partner. During an ongoing negotiation the initiator as well as the responder may lose the partner.

Before a new part of the *IPS state* is set, *potential initiators*, *initiators* and *responders* always check whether the selections of previous decision processes (according to our sequential approach) are still available. The complete dependencies of the sequential approach can be seen in Figure 4.

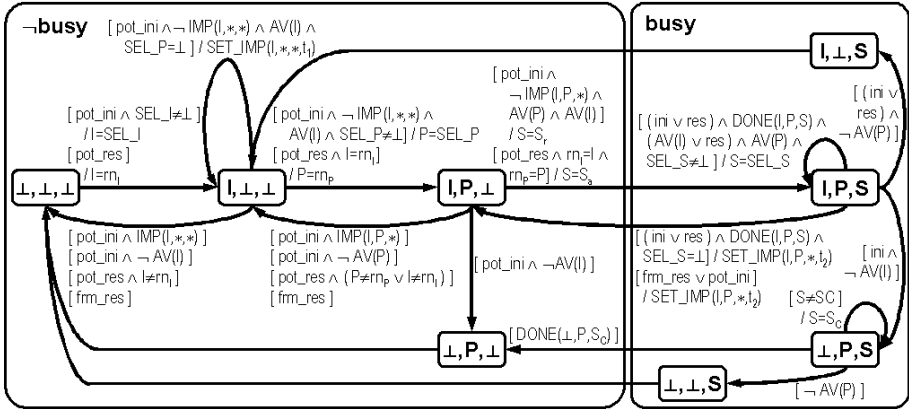


Fig. 4. Extended dynamics for the sequential approach in open environments.

5 Conclusion and Next Steps

In this article we described the specification of interdependencies between different components in a structured reasoning process for negotiating agents. We motivated the application of the C-IPS framework for structuring the reasoning process. The basic dynamics that cover only the primary state transitions were extended by transitions that enable the handling of environmental changes that have not been considered before. For instance, in our extended dynamics we consider an abrupt disappearance of negotiation partners, an unexpected loss of conflict, and a sudden inability to contribute to a conflict resolution.

Despite the C-IPS approach structures the agent's reasoning process, it abstracts from the number of partners or issues under consideration. It is also independent of the agent's characteristics. But C-IPS locates the decisions that might be affected by these parameters. Hence, the framework can serve as a frame for classifications that build on such parameters, e.g. [AWN03]. Altogether, we think that there is no comprehensive framework handling all the questions mentioned so far at an appropriate level of abstraction.

5.1 A New Level of Analysis and Modeling

The C-IPS framework together with IPS dynamics provides a new level of analysis of negotiating agents. It is above the level of negotiation mechanism design, which covers negotiation protocols and strategies. While dependencies between different aspects of the negotiation decision are frequently given only implicit, IPS dynamics make them explicit. The framework's three separate decision components also provide an interesting way of modularization of negotiating agents' internal decision processes. This eases the modeling of more flexible negotiating agents. The modularization also provides a base for exchanging single components of negotiating agents.

5.2 Further Research

Our next steps regarding the C-IPS framework aim at further exploring the advantages and the application of the framework. Because it makes comparison, evaluation, and discussion of different solutions easier we are going to do this for several influential approaches to negotiating agents. Currently, we also use the C-IPS framework for evaluating how different learning strategies can be integrated into negotiating agents. Besides mapping technologies we also map sociological concepts to the decision processes made by negotiating agents.

Because C-IPS widens the horizon, we now start exploring more deeply questions regarding different solutions for conflict identification, issue and partner selection. These solutions may also go beyond individual decisions by applying complex social interaction to establish issues and negotiation groups. Thereby we address more generally the question of agenda setting.

References

- [AWN03] R. Lomuscio A., M. Wooldridge, and R. Jennings N. A classification scheme for negotiation in electronic commerce. *International Journal of Group Decision and Negotiation*, 12(1):31–56, 2003.
- [FWJ03] S. Fatima, M. Wooldridge, and N. R. Jennings. Optimal agendas for multi-issue negotiation. In *Proceedings of 2nd International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 129–136, Melbourne, Australia, 2003.
- [HS99] Michael N. Huhns and Larry M. Stephens. Multiagent systems and societies of agents. In Gerhard Weiss, editor, *Multiagent systems: a modern approach to distributed artificial intelligence*, pages 79–120. MIT Press, 1999.

- [JFL⁺00] N. R. Jennings, P. Faratin, A. R. Lomuscio, S. Parsons, C. Sierra, and M. Wooldridge. Automated negotiation: Prospects, methods and challenges. *Int. Journal of Group Decision and Negotiation*, 2000.
- [KL02] K. Kurbel and I. Loutchko. Multi-agent negotiation under time constraints on an agent-based marketplace for personnel acquisition. In *Proceedings of the 3rd International Symposium on Multi-Agent Systems, Large Complex Systems, and E-Business (MALCEB2002)*, pages 566–579. Erfurt, Germany, October 2002.
- [MUSG03] M. Meister, D. Urbig, K. Schröter, and R. Gerstl. Agents Enacting Social Roles. Balancing Formal Structure and Practical Rationality in MAS Design. TUTS Working Papers TUTS-WP-6-2003, Technische Universität Berlin, Institut für Soziologie, 2003.
- [RZ94] S. Rosenschein, J. and G. Zlotkin. *Rules of encounter: designing conventions for automated negotiation among computers*. The MIT Press, Cambridge, London, 1994.
- [San99] Tuomas W. Sandholm. Distributed rational decision making. In Gerhard Weiss, editor, *Multiagent systems: a modern approach to distributed artificial intelligence*, pages 201–258. MIT Press, 1999.
- [SW03] M. Ströbel and C. Weinhard. The montreal taxonomy for electronic negotiations. *Group Decision and Negotiation*, 12:143–164, 2003.
- [UMS03] D. Urbig, D. Monett Díaz, and K. Schröter. The C-IPS architecture for modelling negotiating social agents. In M. Schillo et al., editor, *Multiagent System Technologies. Proceedings of MATES 2003 (LNAI 2831)*, pages 217–228. Springer, 2003.
- [US04] D. Urbig and K. Schröter. C-IPS approach to negotiating agents: Specifying dynamics interdependencies between issue, partner, and step. In *Proceedings of AAMAS '04*, 2004. poster paper (to appear).