Reflection and Norms: Towards a Method for Dynamic Adaptation for MAS

Ingo J. Timm, Andreas D. Lattner, René Schumann

Information Systems and Simulation, Goethe University Frankfurt
Robert-Mayer-Str. 10, 60325 Frankfurt am Main, Germany
timm|lattner|reschu@cs.uni-frankfurt.de

Abstract. The design of self-organizing systems and particular multiagent systems (MAS) is a non trivial task. On the one hand the particular system should show a dynamic behavior according to its environment, to gain a central advantage of distributed systems, on the other hand it has to act on behalf of its user and the final results have to possess acceptable quality. Especially the quality of the overall system's behavior can become a critical issue, if the subsystems have their own objectives they have to optimize. In this paper we present a methodology that can be integrated into MAS for adapting their behavior allowing local optimization while respecting an acceptable level of the system's global goals.

Keywords. balancing autonomy, multiagent simulation, manufacturing control

1 Introduction

In the last years, a trend towards decentralized designs of decision support and decision taking systems can be observed. In contrast to the centralized systems, it is in practical cases not possible to compute optimal results according to a global objective function. Resulting plans and schedules will be suboptimal [1,2]. Even so decentralized control systems are in the focus of current discussion and research. The reason is that (cf. [3])

– systems with higher flexibility and reliability can be designed and
– decentralized control becomes part of current company organization.

But it turns out to be hard to design systems that on the one hand show a flexible behavior and on the other hand act on behalf of the user and reach acceptable solution quality in comparison with e.g. centralized solution approaches. This is especially the case if the entire systems comprise of subsystems that have a local objective they try to optimize. Note that this is a typical situation if the entire system was developed following the divide and conquer engineering paradigm.

The idea of self-organizing systems have attracted attention by researchers that try to overcome this design problem by allowing the system to self-configure

Dagstuhl Seminar Proceedings 09121
Normative Multi-Agent Systems
http://drops.dagstuhl.de/opus/volltexte/2009/1917

itself to the given environments and given objective functions. In the following we look only at such systems. That is each agent has its local goals and objective function that it tries to optimize while achieving its goals. Moreover the agents form a MAS that have to fulfill global goals and to optimize a global objective function within a dynamic environment. As a consequence of the complex problem and the dynamic environment a flexible solution, e.g. based on a MAS, has advantages to classical centralized optimizing approaches. Norms have been identified to be a valuable method to allow multiagent systems (MAS) flexible behavior still having a hand on the overall system behavior. But the norm design and system configuration respectively calibration can become a complex and time consuming issue. The idea of dynamic adaptation is that the system adjusts itself seamlessly to the given situation [4]. In this paper we outline a method that can be integrated into a MAS that gives the ability of dynamic adaptation respecting local and global goals and objectives. Local entities can perform their actions, and if necessary adapt, enforced or not, their behavior towards the global objective. This step is called strategy revision. If this revision is not sufficient for some reason, a more general scheme is applied. Based on sociologically concepts we propose a reflection phase that allows to change the existing strategy revision strategies or even objectives. Within this method norms can play an important role, as they can not only be used to control the system directly, but can be used guiding the reflection phase as well.

As a step within this methodology we present one possible implementation of the strategy revision step, called regulated autonomy [5]. This is a centralized rule-based implementation of a strategy revision step. For the evaluation of this approach we use a very simple manufacturing scenario. A simple scenario is chosen to provide competitive results of this technology with other classical approaches, that typically cannot be applied to complex problems as they can be found in practice.

This paper is structured as follows. In the next section we outline related work concerning with the notion of autonomy in MAS and the adaption of autonomy at runtime. The related work for norm-based systems is left open for further discussion. In section 3 the principles of reflections are presented. The concept of regulated autonomy, already developed as an efficient implementation of a strategy revision strategy is presented in section 4. Finally we summarize and sketch possible issues to be addressed in future research.

2 Related Work

In this section we review existing work on autonomy and adaption of autonomy in MAS. The special focus of norms is left open for further discussions during the workshop.

2.1 Autonomy in MAS

In the literature, there are discussions on different levels of autonomy [6,7]. The definition of autonomy in the literature ranges from very wide autonomy [8] to

restricted cases [9].
In early multiagent research, Castelfranchi and Conte [8] discuss a very high degree of autonomy, such as the influence of predefined norms, behavior patterns, or procedures is irrelevant, and the relevance is very low with respect to the action-selection process within an agent, respectively.
A more theoretical and interdisciplinary approach to define the term autonomy was presented by Bertschinger et al. [10]. Their key question is how autonomy can be measured. Therefore, they use an information theoretic perspective which bases on the distinction between the system and its environment. Depending on the ability of the system to influence the environment, Bertschinger et al. present different metrics for autonomy.
Additionally, the scope of autonomy – or more precisely the scope or the context autonomy refers to – is discussed by Kirn [11]. Kirn points out that autonomy can address different aspects:

– association autonomy (decision freedom towards the participation of agent societies)
– cooperation autonomy (towards the participation in cooperative processes)
– execution autonomy (local execution)
– resource autonomy (disposition of local resources)
– communication autonomy (this is the participation in communication)

In each of these aspects, the autonomy can be specified for the agent.
Bradshaw et al. [12] define the term autonomy in respect to their research about adjustable autonomy. From their perspective, autonomy is mainly characterized by two aspects:

– self-sufficiency, as the capability of an entity to take care of itself and
– self-directedness, as the freedom from outside control.

According to Bradshaw et al., autonomy can be related to two dimensions. A descriptive dimension which describes if the agent is capable to perform an action and a prescriptive dimension describing if the agent is allowed to perform an action.
Barber and Martin [13] define and measure autonomy as the interdependency of an agent in its decision making to achieve its goals. An agent is autonomous if it is capable to pursue some goals without interference by other agents.
Luck et al. [14] present a strong definition of autonomy. According to them the self-generation of goals is the defining characteristic about autonomy. These goals are generated or derived from motivations an agent has encoded.
Schillo [15,16] introduces a "Framework for self-Organization and Robustness in Multiagent systems" (FORM) where delegation is the main concept in order to describe organizational relationships. He distinguishes between task and social delegation and four different mechanisms for these two delegation modes (Economic Exchange, Gift Exchange, Authority, and Voting). He defines a spectrum of seven organizational forms for groups of agents by using the delegation types and modes as building blocks: Single Autonomous Agents, Task Delegation, Virtual Enterprise, Cooperation, Strategic Network, Group, and Corporation.

Nickles et al. [17] present a specification schema for computational autonomy based on sociological role theory, namely RNS ("Roles, Norms, Sanctions"), in order to "support developers of agent-oriented applications in specifying the kind and level of autonomy (...)". The viewpoint of the authors is that agents act as role owners encountering certain norms in a social frame which regulates the behavior of the agents. In RNS, three types of norms ("permissions, obligations, and interdictions") as well as two sanction types ("reward and punishment") are distinguished [17]. The sanctions can be specified explicitly by the designer and thus provide means to control the autonomy of the agents.
Nevertheless, autonomy is a property, which may lead to partially unwanted system states resulting from conflicting or inconsistent goal sets. The dynamic and complex interdependencies of autonomous subsystems can lead to systems whose organization emerges at runtime. Thus, software engineers of autonomous systems may not consider any possible constellation of subsystems at design time.

2.2 Runtime Adaptation of Autonomy in MAS

The runtime adaptation of autonomy in MAS is addressed by researchers with different application scenarios in mind, thus different terminologies evolved.
From the research about mixed-initiative interactions where agents and humans work together, mostly agents are guided by human operators. In this area the term *adjustable autonomy* has been established. Work about adjustable autonomy can be found, for instance, in [18,15,19,12]. Here we detail the approach by Bradshaw et al. [12]. The goal of adjustable autonomy is to maximize the opportunities for local adaption to unforseen problems and opportunities while assuring humans that agent behavior will be kept in desired bounds. Therefore different adaptations are possible:

- adjusting permission, add or remove rights
- adjusting obligations, add or remove tasks
- adjusting possibilities, add or remove skills
- adjusting capabilities, add or remove resources

These adjustments are done by the human operator, at his will.
With a slight different notion, but with the human-agent interaction focus Urbig and Schröter [20] describe the concept of "dynamic degrees of delegation" from "Full Autonomy to Manual Control". The work is based on the C-IPS approach which addresses different aspects of negotiation decisions in agents (C-IPS stands for external constraints (C), negotiation issues (I), partners (P), and negotiation steps (S)). Basically, agents would act with full autonomy. In order to let the user control the agent, Urbig and Schröter introduce means to distinguish between situations where the user should be involved and situations that can be handled autonomously by the agent. The degree of delegation can be specified for different decision types and dynamically changed during run time.
In the context of robotics, different levels of autonomy are discussed by [21].

Gancet and Lacroix define five levels of autonomy and define for each level which abilities and permissions a robot has with the given autonomy level.
Mailer [22] addresses the area of distributed problem solving where he found that it is useful to dynamically centralize the solving of overlapping subproblems in order to find solutions more quickly. He calls this approach mediation-based as it combines techniques from centralized and decentralized problem solving.
Barber et al. [23] present the concept of "Dynamic Adaptive Autonomy". This allows agents to switch autonomy within a defined spectrum. Thereby according to their notion of autonomy, presented above. The degree of autonomy depends on the goal. Thus for different goals of an agent, it can have different levels of autonomy.

Our approach is also of the second kind allowing a runtime adaptation of autonomy during runtime. An superior entity can force agents to follow certain strategies or to perform actions needed to obtain the desired overall system's performance.

3 Reflection

As mentioned in the introduction, there are challenges in engineering multiagent systems with respect to their properties of autonomy and interaction. The key reasons for applying agent technology in complex economical environments can be found in the high level of modularization and information hiding as well as potential for positive emergent effects. The question arises, if such an emergent behavior, i.e., a macroscopic behavior on the basis of microscopic interactions, is beneficial for the global system. In the beginning of multiagent research, this assumption was stated as a fact. Recent research focuses on sophisticated design of the autonomous subsystems to enable a positive effect of the whole system [24,25]. De Wolf and Holvoet [26] propose an approach for engineering self-organizing systems. Their approach is based on the analysis of the system after implementation and before delivery. Because of the well-known complexity of testing concurrent systems, the approach seems to be adequate for systems with a moderate amount of internal states, where no extensive internal states or static strategic behavior exists. However, sophisticated engineering is required to ensure the desired behavior on the basis of autonomous systems respectively balancing microscopic and macroscopic behavior.

In social science, the phenomena of microscopic-macroscopic interaction is widely researched [27,28,29]. Norms and regulations are introduced in a social system to establish a better system performance. In the following we introduce a methodology which is based on results from social science by Luhmann [30]. The theory focusses on reorganization processes in societies and specifies different steps of individual and social reflection. In this work, a system has the ability to reflect about its overall performance explicitly within negotiations. In an interdisciplinary research [1], we conceptualized a social mechanism as an explanatory model for societies based on the work of [30]. On this basis, a conceptual model for reflection in multiagent systems has been developed.

The multiagent conceptualization of reflection is based on the assumption that a multiagent system was chosen deliberately as a system design. In consequence, autonomy of the agents is not a side effect but one of the key features. If dependability on the multiagent level is in question, then some dynamic mechanism is required, which allows for context-dependent adjustment of the individual behavior. As the autonomy is a key feature, the adjustment of the individual behavior should be as restricted as possible. Furthermore, we aim at a reflection methodology, which preserves local autonomy even by global adjustment.

In economical systems we know these mechanisms for a long time. In the last decades, there is a trend in business administration manage people by delegating tasks with the definition of the context rather than supervising each step of execution. If the context or the boundaries are not met, the management is involved again to handle the exception. Keeping this in mind, the methodology of reflection consists of four steps as illustrated in Figure 1.

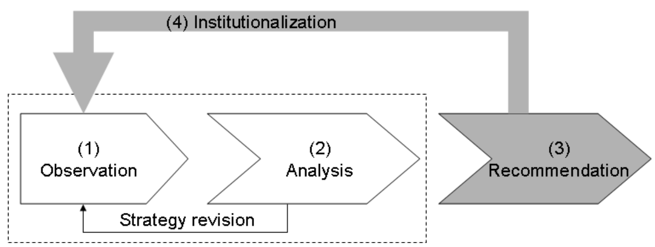

Fig. 1. Reflection in multiagent systems

The process of reflection is divided in two different parts. In the observation and analysis step, the individual agents are reflecting their behavior with respect to the global goals. The step of joint solution and institutionalization includes a group of agents which try to solve a problem cooperatively.

The approach is based on a group of agents which can be formed dynamically in runtime or specified at design-time. In either way, it is assumed, that the group of agents have some common goals and the fulfilment of this goal can be measured by the group or some other instance with communication capabilities. Furthermore, for each goal, there are different levels of goal satisfaction, i.e., 0 implies that a goal is completely unsatisfied and 1 indicates that the goal has been satisfied. For utility-based goals a continuous scale is assumed while for logic-based goals the goal-satisfaction is a binary function. Additionally, we assume that the consideration of global goals in every deliberation step would be inadequate with respect to computational or memory consumption. The basic assumption is that global system goals are known and that it is possible to evaluate a situation to what extent the target setting is achieved, i.e., a system has the ability to reflect about its overall performance explicitly within negotiations. It is distinguished between three color codes: target accomplished ("green"), target slightly failed ("yellow"), and target failed ("red").

The global context emerges from a cooperation or a collaboration. In a cooperation, agents does not have to fulfill the criteria of individual rationality as introduced by Sandholm [31], i.e., agents may "suffer" by a joint solution without compensation. In collaborations, i.e., in settings, where competing agents are cooperating for a specific task, process, or time period, it is important, that the solution strategy is modeled in such a way, that each agent meets individual rationality, i.e., the agents are better off participating in the group than not participating.

In the observation stage, each agent reports its performance to a blackboard or central entity (group coordinator) within the group. The blackboard or the group coordinator computes the goal satisfaction on the basis of the individual results. Together with the concrete satisfaction level, the goal satisfaction is available for any agent of the group. If the goal-satisfaction is classified as deficient, the agents should adjust their operational autonomy. Doing so, the agents should plan the next action or action sequence under consideration of the global goal. E.g., assume that a BDI agent has instantiated an intention and associates this intention with a partial global plan. The agent would now choose a linearization of the plan which is most suitable for supporting the global goal.

The observation stage is used for normal system performance. If the system performance with respect to a specific goal is critical (or cannot be achieved completely for some time, code "yellow"), the multiagent system's state changes to the analysis stage. In this stage, the agents have to communicate their currently pursued goals. The analysis is performed by the agents cooperatively or by a central entity (group manager). The group manager has to identify the interdependencies of the goal selections of the individual agents and missing system performance on the group level. These interdependencies are then published. Under consideration of the autonomy of individual agents, the tactical autonomy has to be adjusted by the agents. Each agent should consider the effects of its goal instantiation, e.g., in our example the step of associating a plan to intentions, with respect to the group performance.

There are situations where an uncoordinated treatment of the mismatch of global goals by individual adaptations cannot lead to satisfying results. This can be the case especially if many agents adapt their behavior in a similar way which can lead to the over-achievement of one goal while the performance decreases w.r.t. other system goals. In the case of a severe system performance, the group of agents is transformed into the joint solution group. Here, the group manager mediates the negotiation about individual agents' goals. The agents are assumed to improve their strategic autonomy, i.e., the agents instantiate those goals which help the group performance.

The solution which has been negotiated in the group and which restored system's performance is generalized as a social rule for later usage in severe situations (phase four), i.e., the experiences of phase three are made persistent for future situations and costly computation and communication can be avoided by handling similar situations in previous phases. For more details about this approach see [1].

4 Regulated Autonomy

As already mentioned the concept of regulated autonomy is an implementation of the strategy revision, shown in figure 1. It is implemented in as a rule-based approach with a centralized entity for monitoring. Typically for rule-based systems the reaction scheme is statically encoded.

The main idea of the concept of regulated autonomy is sketched in figure 2. In default mode, each agent is free to select its behavior as desired (Fig. 2a). Whenever the system performance reaches a critical state, phase two is initiated. In this phase, the manager agent instructs the shop agents to change their strategy in order to improve the system performance (Fig. 2b). Whenever the strategy is changed, costs of the strategy adaptation is recorded. If the system's overall performance reaches an acceptable status, the entities are allowed to use their own local strategies again (Fig. 2c).

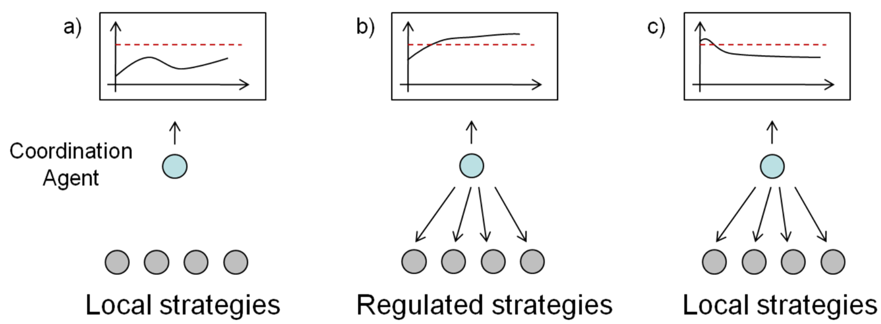

Fig. 2. Autonomy vs. regulation

4.1 Scenario Settings

As in our previous work we use the job shop scenario presented in [32] and [33]. Therefore, we briefly sketch the scenario here. Figure 3 presents a schematic overview of the scenario.

Each shop has an input and an output buffer. It offers exactly one operation. Each job schedules its current jobs using a given dispatching rule. The rules for the shops are assigned randomly from the set of strategies presented in Table 1. These strategies are well known, see e.g. [34].

For simplicity reasons, transportation is not modeled explicitly. It is assumed that enough transport capacity is always available and transportation time is zero. The job characteristics are taken from Brennan and O [33] where the scenario is used as well. In Table 2 the duration and shop sequence are summarized.

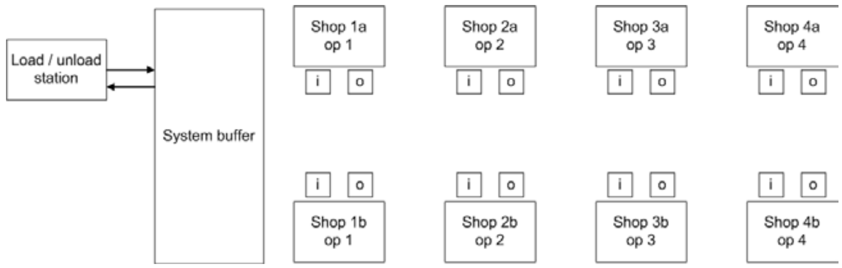

Fig. 3. Shop layout, according to [32]

Strategy Code	Description
SIRO	Service in random order
FIFO	First in first out
SPT	Shortest processing time first
LPT	Longest processing time first
WSPT	Weighted SPT

Table 1. Dispatching strategies for shops

Step / job type	1	2	3	4
J1	6/1	8/2	13/3	5/4
J2	4/1	3/2	8/3	3/4
J3	3/4	6/2	15/1	4/3
J4	5/2	6/1	13/3	4/4
J5	5/1	3/2	8/4	4/3

Table 2. Process plan for different jobs, encoded as time/operation, according to Brennan and O [33]

There exist five different job types which differ in their processing time for each operation and the sequence of operations needed to be performed. The jobs choose the next shop using the shortest queue strategy.

As an objective function for the overall system we use the mean flow time. This implies that, regarding the given dispatching rules, it is known that the SPT dispatching rule will perform best. This eases the application of a rule-based implementation of the strategy revision process. E.g., if another objective function is used to minimize the makespan, the optimal distribution of dispatching rules depends on the set of orders, and has to be computed for each revision process.

As already mentioned there exists one entity, called manager agent, that supervises the overall performance. If a job is finished, the corresponding job agent informs the manager and reports the flow time of this job. The manager agent can monitor the mean flow time according to the jobs finished so far. If this value falls below a specified threshold, the manager agent can order the shop agents to work following a specific strategy. Here this is the SPT strategy which is known to perform best in this scenario. If the actual mean flow time reaches an acceptable range again, it can allow the shops to work according to their locally preferred strategy.

4.2 Evaluation

The results presented in this section were computed using a time driven simulation implemented as a multiagent system based on the JAVA agent development framework JADE[1].

For evaluation purposes, we use three basic settings with different control cycles w.r.t. the overall system's performance. In the `Con01` experiments, the current quality (mean flow time) is checked after each job. In the `Con03` and `Con10` settings, the control interval is set to three and ten, respectively. For each setting, ten different runs are performed where 100 jobs are generated and processed. Figure 4 shows the average mean flow times for all three settings[2]. The mean flow time is computed every time a job has been finished, i.e., the last value (job no. 100) indicates the mean flow time of all 100 jobs.

Figure 5 presents Box-Whisker plots of the relative central control time, i.e., the ratio of time interval lengths under central control divided by the total time. Box-Whisker presents the upper and lower quartile and the median. Therefore, they can be used to discuss the statistically spread of the data. While the mean central control times of setting `Con01` and `Con03` do not differ a lot, the mean value of `Con10` is lower indicating that in our experiments central control is rather infrequent. Having in mind that these regulated strategies are capable to ensure an adequate level of the overall performance (see [5]) it can be stated, that this can be done restricting the local autonomy rarely.

[1] For the Java Agent DEvelopment Framework see `http://jade.tilab.com/`.

[2] All statistical computations as well as plots have been generated with R Project for Statistical Computing 2.6.1, see `http://www.r-project.org/`.

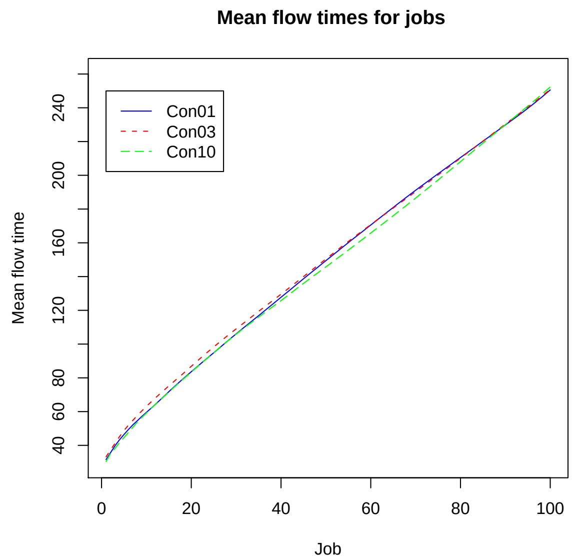

Fig. 4. Mean flow times

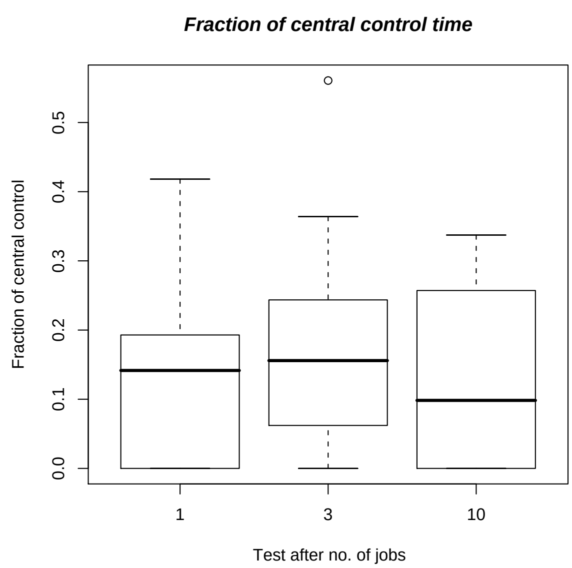

Fig. 5. Box-Whisker plot of relative central control times

5 Reflection and Norms

So far, we presented a methodology for the dynamic adaptation of multiagent systems. The concept itself is – deliberately – on a rather abstract level such that a huge variety of methods for concrete dynamic system design can be derived. Thus, the presented methodology of dynamic adaptation should be rather seen as a building block for software systems; we explicitly do not propose an software architecture. The dynamic adaptation, i.e., the strategic management, of autonomous software systems constitutes the center of our research agenda. Until now we have investigated three aspects. As a first step we[3] have designed the overall methodology as an interdisciplinary model with a mapping of Luhmann's concept of reflection following to strategic management of autonomous software systems. In a second step, we[4] investigated the feasibility of observation and analysis of interdependencies between local and global objectives. In this context, we applied objectives on the basis of key performance indicators and utility functions; the assessment of dependencies are derived automatically. The third aspect, the implementation of the strategy revision process, has been done as a statical rule-based approach (regulated autonomy) which has been described briefly in previous section.

We rely on mechanisms adapted from social science. To stay in line with this research and the terminology, the use of norms seems promising. Having in mind the abstract concept mentioned above, there are no restrictions for the formalism of norms to be applied.

To our point of view, it is promising to use norms for the implementation of the following aspects of the reflection methodology. As already mentioned, the institutionalization phase of the reflection can be implemented as norms to conserve successful strategies persistently.

In our case study (regulated autonomy) we expect high potential of substituting the static rules by notions of norms.

The most challenging process step of our methodology is the generation joint solutions in the recommendation phase. Various approaches like central decision making, argumentation, negotiation are applicable. We are convinced that none of the approaches is dominant with respect to different application domains. Consequently, the process of finding joint solutions should be guided by norms.

Until now the discussion focussed on how to apply norms on the process of reflection. However, we assume that there is high potential for applying our methodology of dynamic adaptation to the evolution of norms. From a more general perspective, this methodology can be used to enrich a model for social simulation.

We are looking forward to discuss these issues in Dagstuhl.

---

[3] In cooperation with the social scientist Frank Hillebrandt (University of Muenster).

[4] Together with the diploma student Florian Pantke (University of Bremen).

References

1. Timm, I.J., Hillebrandt, F.: Reflexion als sozialer Mechanismus zum strategischen Management autonomer Softwaresysteme. In Schmitt, M., Hillebrandt, F., Florian, M., eds.: Reflexive soziale Mechanismen. VS Verlag für Sozialwissenschaften, Wiesbaden (2006)
2. Schumann, R., Sauer, J.: Implications and consequences of mass customization on manufacturing control. In Blecker, T., Edwards, K., Friedrich, G., Salvador, F., eds.: IMCM'07 + PETO'07. Volume 3 of Series on Business Informatics and Application Systems., Hamburg, GITO (2007) 365 – 378
3. Kirn, S., Herzog, O., Lockemann, P., Spaniol, O.: Multiagent Engineering. International Handbooks on Information Systems. Springer, Berlin et al. (2006)
4. Kephart, J.O., Chess, D.M.: The vision of autonomic computing. Computer **36** (2003) 41–50
5. Schumann, R., Lattner, A.D., Timm, I.J.: Regulated autonomy: A case study. In Mönch, L., Pankratz, G., eds.: Proceedings of the Conference Track "Intelligente Systeme zur Entscheidungsunterstützung" at the Multikonferenz Wirtschaftsinformatik (MKWI 2008), Munich, Germany, SCS Publishing House (2008) 83–98
6. Falcone, R., Castelfranchi, C.: Grounding autonomy adjustment on delegation and trust theory. Journal of Experimental and Theoretical Artificial Intelligence **12** (2000) 149–151
7. Rovatsos, M., Weiss, G.: Autonomous software. In Chang, S.K., ed.: Handbook of Software Engineering and Knowledge Engineering. Volume 3: Recent Advances., River Edge, New Jersey, World Scientific Publishing (2005)
8. Castelfranchi, C., Conte, R.: Emergent functionality among intelligent systems: Cooperation within and without minds. Journal on Artificial Intelligence and Society **6** (1992) 78–87
9. Steenhuisen, J.R., Witteveen, C., ter Mors, A.W., Valk, J.M.: Framework and complexity results for coordinating non-cooperative planning agents. In Fischer, K., Timm, I., André, E., Zhong, N., eds.: Multiagent System Technologies (MATES). LNAI, Erfurt, Springer Verlag, Berlin, Heidelberg, New York (2006) 98 – 109
10. Bertschinger, N., Olbrich, E., Ay, N., Jost, J.: Autonomy: An information-theoretic perspective. Biosystems (2008) 331–345
11. Kirn, S.: Kooperierende intelligente Agenten in Virtuellen Organisationen. HMD **185** (1995) 24 – 36
12. Bradshaw, J.M., Feltovich, P.J., Jung, H., Kuklarni, S., Taysom, W., Uszok, A.: Dimensions of adjustable autonomy and mixed-initiative-interaction. In Nickles, M., Rovatsos, M., Weiss, G., eds.: AUTONOMY 2003 : International Workshop on computational autonomy. Volume 2969 of LNAI., Melbourne, Springer, Berlin (2004) 17 – 39
13. Barber, K.S., Martin, C.E.: Agent autonomy: Specification, measurement, and dynamic adjustment. In: Proceedings of the Autonomy Control Software Workshop at Autonomous Agents 1999 (Agents99), Seattle, WA (1999)
14. Luck, M., d'Inverno, M., Munroe, S.: Autonomy: Variable and generative. In Hexmoor, H., Falcone, R., Castelfranchi, C., eds.: Agent Autonomy. Multiagent Systems, Artificial Societies, and Simulated Organizations. Springer (2003) 9 – 22
15. Schillo, M.: Self-organization and adjustable autonomy: two sides of the same coin? Connection Science **14** (2002) 345–359
16. Schillo, M.: Self-organization and adjustable autonomy: Two sides of the same medal? In Hexmoor, H., Falcone, R., eds.: Proceedings of the AAAI2002 Workshop

on Autonomy, Delegation, and Control: From Inter-agent to Groups, Menlo Park, CA, AAAI Press (2002) 64–71
17. Nickles, M., Rovatsos, M., Weiss, G.: A schema for specifying computational autonomy. In: Proceedings of the Third International Workshop "Engineering Societies in the Agents World" (ESAW'2002), Madrid, Spain (2002)
18. Schurr, N., Marecki, J., Tambe, M.: Riaact: A robust approach to adjustable autonomy for human-multiagent teams. In Padgham, L., Parkes, D.C., Müller, J.P., Parsons, S., eds.: Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008), Estoril, Portugal (2008) 1429–1432
19. Scerri, P., Sycara, K., Tambe, M.: Adjustable autonomy in the context of coordination. In: AIAA 3rd "Unmanned Unlimited" Technical Conference, Workshop and Exhibit. (2004)
20. Urbig, D., Schröter, K.: From full autonomy to manual control. In: Proceedings of the Workshop Concurrency, Specification & Programming (CS&P'05), Ruciae-Nida, Poland (2005) 520–531
21. Gancet, J., Lacroix, S.: Embedding heterogeneous levels of decisional autonomy in multi-robot systems. In: 7th International Symposium on Distributed Autonomous Robotic Systems, Toulouse (France) (2004)
22. Mailler, R.T.: A mediaton-based approach to cooperative, distributed problem solving. PhD thesis, University of Massachusetts Amherst (2004)
23. Barber, K.S., Goel, A., Martin, C.E.: Dynamic adaptive autonomy in multi-agent systems. Journal of Experimental and Theoretical Artificial Intelligence **12** (2000) 129–147
24. Liu, J.: Self-Organized Autonomy in Multi-Agent Systems. In: Autonomous Agents and Multiagent Systems - Explorations in Learning, Self-Organization and Adaptive Computing. World Scientific, Singapore (2001) 141–180
25. Liang, H.H., Zhu, M.L.: Developing self-organized architecture solution according to model driven generative domain engineering. In Czap, H., Unland, R., Branki, C., Tianfield, H., eds.: Self-Organization and Autonomic Informatics (I). Volume 135 of Frontiers in Artificial Intelligence and Application., Amsterdam, IOS Press (2005) 97–104
26. de Wolf, T., Holvoet, T.: Towards a methodology for engineering self-organising emergent systems. In Czap, H., Unland, R., Branki, C., Tianfield, H., eds.: Self-Organization and Autonomic Informatics (I). Volume 135 of Frontiers in Artificial Intelligence and Application., Amsterdam, IOS Press (2005) 18–34
27. Schegloff, E.A.: Between macro and micro: Contexts and other connections. In Alexander, J., Giesen, B., Münch, R., eds.: The Micro-Macro Link. University of California Press, Berkeley (1987) 207–237
28. Bohman, J.: The Macro-Micro Relation. In: New Philosophy of Social Science - Problems of Indeterminancy. The MIT Press, Cambridge (1993) 146–185
29. Alexander, J.C., Giesen, B.: From reduction to linkage: The long view of the micro-macro link. In Alexander, J., Giesen, B., Münch, R., eds.: The Micro-Macro Link. University of California Press, Berkeley (1987) 1–45
30. Luhmann, N.: Soziale Systeme, Grundrisse einer allgemeinen Theorie. Suhrkamp, Frankfurt (1984)
31. Sandholm, T.: Distributed rational decision making. In Weiss, G., ed.: Multiagent Systems: a modern approach to distributed artificial intelligence. MIT Press (1999) 201– 258
32. Cavalieri, S., Bongaerts, L., Macchi, M., Taisch, M., Weyns, J.: A benchmark framework for manufacturing control. In: Second International Workshop on Intelligent Manufacturing Systems, Leuven, Belgium (1999) 225 – 236

33. Brennan, R.W., O, W.: A simulation test-bed to evaluate multi-agent control of manufacturing systems. In: WSC '00: Proceedings of the 32nd Conference on Winter Simulation, Orlando, Florida, Society for Computer Simulation International (2000) 1747–1756
34. Pinedo, M.: Scheduling: Theory, Algorithms and Systems. Industrial and Systems Engineering. Prentice-Hall (1995)