Published in: Hülsmann, M.; Windt, K. (eds.): Understanding Autonomous Cooperation & Control in Logistics – The Impact on Management, Information and Communication and Material Flow. Springer, Berlin, 2007

Autonomy in Software Systems

Ingo J. Timm[1], Peter Knirsch[2], Hans-Jörg Kreowski[2], Andreas Timm-Giel[3]

[1]Faculty of Computer Science and Mathematics, Johann Wolfgang Goethe-Universität Frankfurt, Frankfurt am Main, Germany
[2]Faculty for Mathematics and Computer Science, University of Bremen, Bremen, Germany
[3]Faculty for Physics and Electrical Engineering/Information Technology, University of Bremen, Bremen, Germany

Introduction

Looking at the whole logistic network, the structure of logistic processes becomes increasingly complex. Especially in transport logistics, atomisation of transportation processes, multimodal transport chains, international competition, changing ecological and legal constraints along with congestion of traffic infrastructure lead to highly dynamic and complex logistic processes that are difficult to plan (in advance).
The same situation can be found in production logistics. Modern production processes allow highly customized products. But the need for the reduction of costs, emerging virtual enterprises with distributed production plants, and just-in-time production leads to complex and highly dynamic production processes again being difficult to plan.
The described complexity and the arising difficulty in planning is a great challenge for enterprises. Having means to overcome these problems can constitute a significant competitive advantage.

The vision of the Collaborative Research Centre CRC 637 *Cooperating Logistic Processes* is to equip logistic processes and logistic objects with the capability to take decisions autonomously based on local and partially

incomplete information. In consequence, the necessity to plan on a high level of details should be reduced significantly.
Considering transport logistics, this means that transport goods, transport vehicles etc. take decisions, like using a different route because of traffic congestion, locally without reinitiating a new overall planning and optimisation process. Similarly in production logistics, intelligent goods can select different suitable tools for the next production step.

Within the CRC 637 the autonomy in logistic systems and its benefits are investigated from different disciplines. To support autonomy, logistic entities need to have a minimum intelligence. Transport goods need to have some means of interaction, communication, and processing capabilities to take decisions, act, interact and communicate, autonomously. Logistic Systems are distributed and integrate physically mobile entities like transport vehicles or transport goods.
The CRC 637 is developing integrated solutions and management strategies using recent technological advances, on the hardware as well as on the software side, e.g., RFID, WLAN, agent technology.

This chapter presents autonomy as a core property of innovative software systems like agents and autonomous units. In the first section, ideas of agency are introduced. In the following section, autonomous units as a graph transformation-based approach to handling autonomous decision makers in a formal framework are compared with agents (for a detailed introduction of autonomous units the reader is referred to Chapter XXX). Finally, advanced concepts of agency are discussed.

Ideas of Agency

Since the early 1970s there are various approaches in computer science to design and develop distributed systems to overcome limited computational capacity of single processing units and solving larger problems. Accompanying analysis of other research fields especially, in biology seem to content promising approaches of simple distributed decision entities leading to an emergent somehow intelligent behavior like a human brain. But even simpler contexts, like ant colonies show emergent behavior including "intelligent" solutions resp. global optimization by local interaction.

Let us consider a simple logistics task, which is performed efficiently and reliable by real-world ant colonies: finding the shortest path from the nest to the food source. After random walks in the environment, the ants will identify a new food source. Shortly after the identification, the ants will travel between nest and food on the direct and shortest path (cf. Figure 1 (a)). This solution is reliable with respect to environmental changes. If an obstacle is preventing the use of the shortest path (cf. Figure 1 (b)), the ants will travel around obstacle randomly (cf. Figure 1 (c)). Again, after short time, the ants will take the shortest path. The underlying algorithm is simple with respect to requirements for coordination effort between the ants and computational complexity within an ant. The ants are using pheromones to mark their path. The intensity of the pheromones decreases continuously over time. Ants traveling to the food source follow the path with the strongest pheromone concentration.

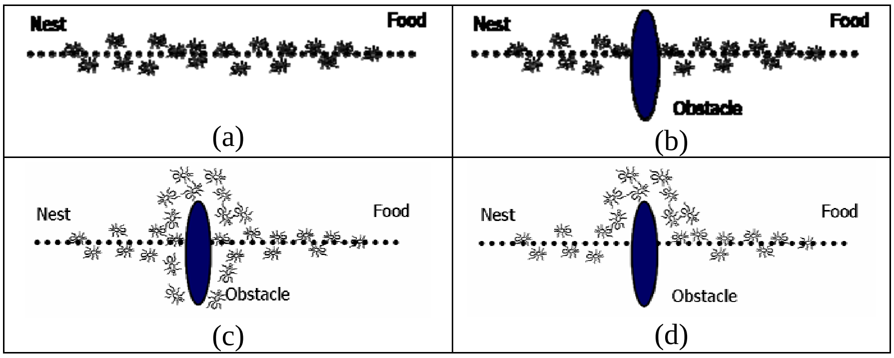


Fig. 1. Ant colony is finding the shortest path.

There are several similar examples in natures, where simple decision entities, like ants, bees, birds, termites, are performing complex problem solving by local interaction.

However, logistics tasks in real-world applications are far more complex due to various partially conflicting objectives, competitive behavior of the entities, etc. Thus, central planning in advance causes exponential computational complexity. Additionally, central planning is often prohibited by competing organizational substructures. Consequently significant research is focusing on emergent systems, where global optimization on a macro-level emerges from local interaction on a micro-level. The underlying idea is, to design autonomous entities, which implement simple decision behavior, which gain complexity by interaction with other autonomous entities.

The concept of autonomies entities interacting on a local level has been researched in computer science since the early 1980s. Smith (Smith 1980) invented the contract net approach to negotiate distributed solution in a system consisting of multiple autonomous decision makers with heterogeneous capabilities resp. skills (Smith and Davis, 1988). The actor theory developed important theoretical models for message-based communication of autonomous entities (actors) (Agha 1986). Following developments constituted the research field on autonomous agents and multiagent systems.

There are several classes of agent technology. A widely accepted definition of agents is provided by Pattie Maes: “Agents are software entities that assist people and act on their behalf” (Maes 1994). For agents in logistics, we propose a specialized definition as follows: Agents are situated in an environment, act autonomously, and are able to sense and to react to changes (Knirsch and Timm 1999).

Autonomous agents are modeled as completely free to negotiate and establish any sort of commitment with any other agent (Müller 1996). Following Castelfranchi and Conte (Castelfranchi and Conte 1992), preexisting norms, habits, and procedures are not relevant for the agents’ actions. Thus social action is explained only in terms of the agents’ mental states as beliefs and intentions. This approach describes the extreme situation of a totally autonomous agent, while in practice partial autonomy is common. This leads to a generalized definition: An agent is autonomous to the extent that its action choices depend on its own experience, rather than on knowledge of the environment that has been built-in by the designer (Russell and Norvig 1994).

From an external view, a system may be defined as autonomous, if it is acting non-deterministically, i.e., the system may function differently in identical situations. However, this does not mean, that an autonomous system has to be non-deterministic. The appearance of non-determinism arises from the limited view on the environmental state (situation). If the internal state of the system is included, an autonomous system might also be deterministic.

A more sophisticated approach to define autonomy resp. autonomous systems is the consideration of properties as introduced in (Timm 2006). In this context, autonomy resp. autonomous agents are best described by the three properties: pro-activity, interaction, and emergence. Pro-activity means, that the agent activates goals resp. initiates actions without specific external events. Therefore, the agent requires the ability to reason about its goals and the current situation, i.e., an explicit representation of goals and environment is required. A main feature of an autonomous agent is the capability of interaction with its environment and other agents. Pro-activity and interaction of agents in multiagent systems cause emerging properties which are not explicitly modeled in advance. The naive formulation of this fundamental assumption is that the system is more than the sum of its parts.

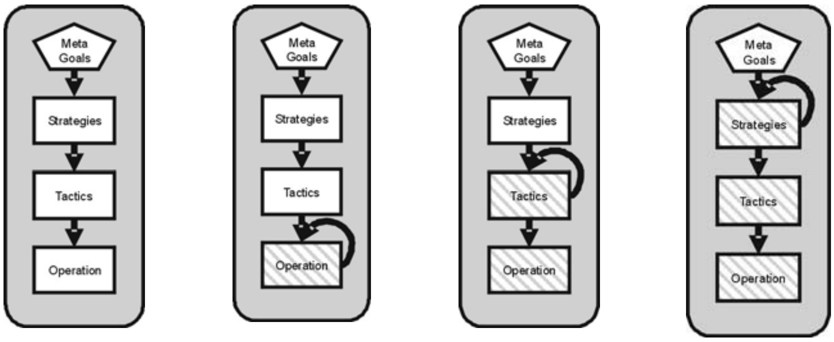


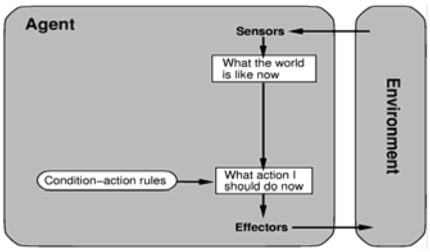
Fig. 2. Levels of autonomy

While Caselfranchi and Conte discuss a very high degree of autonomy, different levels of autonomy are introduced, e.g., (Rovatsos and Weiss 2005), (Müller 1997), (Timm 2006). Russell and Norvig classify the environment to differentiate AI approaches (Russell and Norvig 1994, 2003) following the criteria of observable, deterministic, episodic, static, discrete, and agent-oriented environments. In (Timm 2006), a classification scheme for levels of autonomy is introduced (cf. Figure 2): strong regulation (no autonomy), operational autonomy (reactive systems), tactical autonomy (classical deliberative approaches), and strategic autonomy (complex intelligent systems). Table 1 yields the mapping of levels of autonomy to the environmental properties of Russell and Norvig.

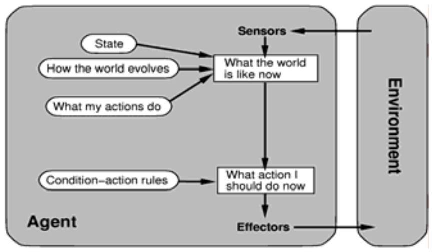
Level of Autonomy	Observable	Deterministic	Episodic	Static	Agents
Strong Regulation	Fully	Deterministic	Episodic	Static	Single
Operational Autonomy	Partial	Deterministic	Episodic	Static	Multi
Tactical Autonomy	Partial	Stochastic	Episodic	Semi	Multi
Strategic Autonomy	Partial	Stochastic	Sequential	Dynamic	Multi

Tab. 1. Classification scheme for levels of autonomy

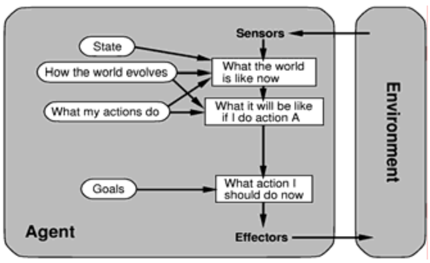
For practical applications or theoretical research a specific architecture has to be developed. The following paragraphs discuss agent architectures as introduced by Russell and Norvig (2003, see Figure 3) in context of the classification scheme for levels of autonomy. In a first step, an agent can be described by its input/output relations (black-box principle, Müller 1996). Russell and Norvig define this approach as the simple reflex agent (cf. Figure 3a), which implements strong regulation with respect to the levels of autonomy. Introducing an internal state and reflection about environmental changes and action consequences combined with condition-action rules lead to operational autonomous systems (cf. Figure 3b). For tactical autonomy it is necessary to deliberate on different objectives; Russell and Norvig suggest that utility-based agents select their goals with respect to the greatest happiness specified by a utility function (cf. Figure 3c). Finally, the strategic autonomy includes deliberation capabilities on goals, plans, and actions (cf. Figure 3d).



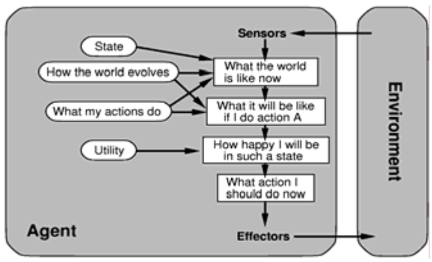
(a) simple reflex agent



(b) model-based reflex agent



(c) utility-based agent



(d) goal-based agent

Fig. 3. Agent architectures (Russell and Norvig, 2003, p. 47, p. 49, p. 52, p. 50)

A unified approach to specify architectures in agent technology is the formal specification with (multi-)modal logics. Wooldridge and Lomuscio invented a general framework for the definition of agents as well as multi-agent systems, which may be outlined as follows (cf. Figure 4, Wooldridge and Lomuscio 2000). The agent behavior may be based on three phases: perceive, next, do.

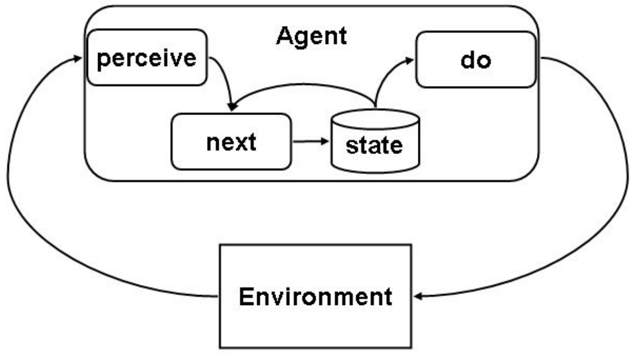


Fig. 4. Agent architecture in VSK (Lomuscio and Wooldridge, p. 3)

1. perceive

For each agent *agt*, there exists a unique environment, which is principally visible for it. Agents observe their environment via sensors in order to identify the relevant information constituting its perceptions:

$$perceive: E \rightarrow P \tag{1}$$

where E is the set of environmental states and P is the set of perceptible information.

2. next

Depending on the agent's internal architecture and state design, it is able to deliberate, plan, or select appropriate actions for execution. Let L denote the set of local states of agent *agt*. The reasoning process may be formalized by:

$$next: L \times P \rightarrow L \tag{2}$$

The local state of an agent may be constituted by highly complex structures. There are several aspects, which have been discussed in order to specify this structure.

3. do

In the third step the agent is selecting an action according to its internal state, which is performed in its environment:

$$do: L \rightarrow Act \tag{3}$$

where Act is the set of possible local actions.

Summarizing an agent can be defined as a system, consisting of the three decision functions *perceive, next, do*, as well as the accompanying concepts environment, perceptions, local states (including an initial state), and actions.

Definition 1

Given a set E of environmental states, an agent is a system $agt = \langle L, P, Act, perceive, next, do, l_0 \rangle$ where

- L is the set of possible agent's states,
- P is a set of perceptible information,
- Act is a set of actions,
- $perceive\colon E \rightarrow P$ is a function for perceiving environmental states,
- $next\colon L \times P \rightarrow L$ a local state transformer function,
- $do\colon L \rightarrow Act$ an action selection function, and
- $l_0 \in L$ the initial state of the agent.

This is a slight modification of the notion of agents of the VSK model. In the original definition, *perceive* is the sequential composition of a visibility function and a see function. The agent environment provides the visibility function for each agent, specifying which parts of the environment are generally perceivable to the agent. The see function belongs to the agent instead of the perceive function. To motivate this separation, let us consider an agent in the Internet as an example. The agent may perceive any web page which is generally accessible, but none which is restricted. Thus the environment "hides" the restricted information to the agent. However, the sensors of the agents may further restrict perceivable information. If an agent does not support a specific protocol for connecting to a web service, the agent may not perceive data provided by the web service, even if the information source is not hidden from the environment.
The visibility function of the environment implements general accessibility to the environment while the see function of an agent maps from environmental state to internal perception representation. However, the separation of the perceive function into two component is not needed in this paper. Hence we have integrated the visibility component into the notion of agents. Therefore, the agent environment is described in a formal sense by Definition 2 as follows.

Definition 2

An environnent is a system $Env = \langle E, Act_1, ..., Act_n, \tau, e_0 \rangle$ where

- E is the set of all possible environmental states,
- Act_i is the set of actions for each $i=1, ..., n$,
- $\tau: E \times Act_1 \times ... \times Act_n \rightarrow E$ is a state transformer function, and
- $e_0 \in E$ is the initial state of the environment.

The formal notion of multiagent systems is given in Definition 3. It combines a group of agents $agt_1, ..., agt_n$ with an environment in such a way that the set of actions of the environment coincides with the agents' sets of actions and all agents perceive the environmental states of the environment.

Definition 3

A multiagent system $MAS = \langle Env, agt_1, ..., agt_n \rangle$ consists of an environment $Env = \langle E, Act_1, ..., Act_n, \tau, e_0 \rangle$ and a sequence of agents $agt_i = \langle L_i, P_i, Act_i, perceive^i, next^i, do^i, l_0^i \rangle$ with, $perceive^i: E \rightarrow P_i$ for each $i=1, ..., n$.

The previous definitions are introducing a static structure of the multiagent systems. During runtime, the agents as well as the environment changes with respect to their states. In the approach of VSK, there is a specified starting point in the systems, constituted by the initial states of the agents as well as the initial state of the environment. Essential property of the multiagent system is the function which gains the current state out of prior states. This system dynamics is outlined in Definition 4.

Definition 4

The *system dynamics* of a multiagent system $MAS = \langle Env, agt_1, ..., agt_n \rangle$ is given by sequences of global environmental states $g_0, g_1, ..., g_t, g_{t+1}, ...$ where the initial global state is defined as

$$g_0 = \langle e_0, next^1(l_0^1, perceive^1(e_0)), ..., next^n(l_0^n, perceive^n(e_0)) \rangle$$

and, given the global environmental state $g_t = \langle e_t, l^1_t, ..., l^n_t \rangle$ with $t \in I\!N$, the next global environmental state is defined as

$$g_{t+1} = \langle \tau (e_t, do^1(l^1_t), ..., do^n(l^n_t)), l^1_{t+1}, ..., l^n_{t+1} \rangle \text{ with}$$
$$l^i_{t+1} = next^i(l^i_t, perceive^i(e_t)) \text{ for } i \in \{1, ..., n\}.$$

Ideas of Autonomous Units

In the following sections, the concepts of *autonomous units* and *communities of autonomous units* as they are introduced in the Chapter "Autonomous Units: Basic Concepts and Semantic Foundation" in this book are compared to the framework of agents and multiagent systems. As a short repetition, an autonomous unit (see also (Hölscher et al. 2006)) is a new, formal, and general modelling concept especially designed for the modelling of autonomous behaviors. An autonomous unit has a goal, a certain set of capabilities, and an internal and therefore autonomous control.
Up to now existing modelling approaches do not cover the topic of autonomous control that explicitly while preserving a level of formality of the description that allows defining a precise semantics and proving certain properties of the system. Autonomous units are an extension of the well-studied and proven to be useful transformation units (see, e.g. (Kuske 2000)) which provide a general structuring methodology for rule-based graph transformation systems but only with a sequential semantics. That means that actors could only perform actions one after the other which is not suitable for logistic processes that are characterised by independent actors performing their tasks independently in a not predefined order and even concurrently. The framework of graph transformation as for instance described in (Ehrig et al. 1999) or in (Janssens et al. 2005) allows to model different kinds of semantics ranging from strictly sequential to concurrent behaviour. Autonomous units -which are still under development - constitute an adequate means for modelling complex networks of independent actors in a structured and rule-based way with an explicit representation of autonomy.
Several similarities with multiagent systems make it worth to have a closer look at the used concepts and their relations.

Relationship between Autonomous Units and Agents

The relationship between autonomous units and agents is discussed with respect to the environmental states, the transformation steps, the perception, and the decision making.

Environmental States

Both approaches assume environments in which agents and autonomous units, resp., act and interact. While the environmental states of multiagent systems are not restricted in any way, the information structures underlying communities of autonomous units are assumed to be graphs. If one chooses a particular kind of graphs, it provides some explicit knowledge about the environmental states how they may be manipulated and how they may be visualized for example. One may say that graphs are particular models of environmental states of multiagent systems. But one should notice that graphs are very generic and flexible structures and that many data structures and system states are easily and adequately represented by graphs. Hence the choice of graphs is not much of a restriction.

Transformation Steps

With respect to the notion of transformation steps, the relation between multiagent systems and communities of autonomous units is similar. A multiagent system assumes some state transformer function of environmental states dependent on an action performed by each agent, i.e. a function

$$\tau : E \times Act_1 \times \ldots \times Act_n \rightarrow E.$$

It is not specified how an environmental state changes under which actions in the general framework, but must be instantiated in each case of application.
In contrast to this, a transformation step in a community of autonomous units is defined explicitly by a direct derivation, i.e. by an application of a rule to an environment graph yielding another environment graph.

$$G \Rightarrow_r H.$$

This provides a particular choice of the environment transformation τ if r is a parallel rule composed of one rule for each autonomous unit. And graph transformation turns out to a model of multiagent systems in this respect.
In another respect, communities of autonomous units have a more general environment transformation than multiagent systems. In addition to the synchronized parallelism of a rule per unit, any kind of sequential or parallel rule may be applied. An autonomous unit can act alone, some – but not necessarily all – of the units may act together, and each of the acting units may apply several rules in parallel in each step. Moreover, there is a concurrent semantics of communities of autonomous units in which synchronized parallelism does not appear explicitly and actions are only ordered in time if they are causally dependent.

Perception

Each agent of a multiagent system has got its individual perception of the environmental states given by the function *vis:* $E \rightarrow 2^E$ and *see:* $2^E \rightarrow P$. As they are always applied together, first *vis* then *see*, they may be replaced by a single function *perceive:* $E \rightarrow P$ given by *perceive(e) = see(vis(e))* for all $e \in E$.

An autonomous unit is not equipped with an explicit perception. Nevertheless, there is a counterpart implicit in the approach. Considering the rules of an autonomous unit, they can access an environment graph G by all possible rule applications. In this sense, the set of all direct derivations $G \Rightarrow_r G'$ with rules of the unit is the perception of G.

Depending on the control condition of the unit, the perception of an environment may contain further information. If the control mechanism is based on an evaluation function, for example, then the perception is enlarged by the view the evaluation provides of the environment. But quite often control conditions are used that check only the possible rule applications. In these cases, the control component of a unit does not add anything to the perception.

Decision Making

Based on the perception of the actual environmental state e, an agent updates its actual local state l by applying the function *next:* $L \times P \rightarrow L$ yielding the next local state, i.e. $l' = next\ (l,\ perceive\ (e))$.
Then it decides about the next action to be performed by applying the function *do:* $L \rightarrow Act$ yielding *do(l')*.

In the framework of autonomous units, this task is done by the control condition of a unit. The control condition checks all possible rule applications of a unit to each actual environment graph and divides them into admitted and forbidden ones. Then one of the admitted rule applications is picked for the next action of the unit. Therefore, the decision of the next action is based on the perception of the environment graph and its restriction to the admitted part, which may be seen as the local state.

Advanced Concepts of Agency

In the section on ideas of agency, fundamental concepts for agency have been discussed following the VSK specification of Wooldridge and Lomuscio (2000). The basic model of the agents is quite simple, defining a perception (*perceive*), state transformation (*next*), and action (*do*) function. The formal representation of architectures and decision behavior has a strong history in the agent community. Agent's formalization mainly depend the constitution of a suitable formal language. The choice and development of the language depends on the use for internal specification used by agents for reasoning about behavior and actions, external specification used by agents for communicating with other agents, i.e., exchanging pieces of knowledge, or external use on a meta-level by developers for specifying, implementing, validating, and verifying properties of agents' behavior.

Internal specification languages are mainly applied to agents, which implement reasoning capabilities for advanced decision behavior. The formal language is used for the representation of the environment or internal state of the agent. Agents using formal languages for reasoning about knowledge to identify an action or action sequences are referred to as intelligent, deliberative, cognitive or rational. Interaction between agents uses communication languages. These languages specify the process of communication and a mandatory syntax of messages. However, message content is not specified there. The specification of content uses an external language, e.g., OWL (Patel-Schneider et al. 2007). Important aspects of external specification languages are that content can be interpreted in the same way by sender and receiver. The complexity of these languages as well as the underlying coordination mechanism vary, e.g., in market-based coordination models, content languages will consist of simple concepts for price and objects, while in negotiation-based coordination models, logical expressions are exchanged and used explicitly for internal reasoning.

Formal specification with meta-language should enable the design of multiagent systems as well as verification and validation of agents' or multiagents' behavior (Dunne et al. 2003). The language VSK is designed for these purposes. However, the individual agent's should be allowed to use varying formal languages for internal reasoning (Singh et al. 1999). The distinction between specification and implementation languages is not only useful for flexibility but also for expressiveness and efficiency. Designers tend to use a formal language with high expressive power for describing an intended system's behavior. In contrast to this, an implementation is in need of computationally efficient realizations, which – at least – rule out those formal approaches which are not decidable.

Modeling heterogeneous multiagent systems requires the abstraction of individual agents' behavior. The model of the system should only include those actions, which are perceivable to other agents or which change the environment. (Wooldridge and Lomuscio 2000) introduce VSK as a formal model for multiagent system based on multimodal logic. VSK integrates an environment depending visibility function (visibility) and an agent depending perception function (see). These concepts realized as modalities enable varying virtual environments for specific agents. A third modality is used for representation of the local state of agents (knowledge). However, the interaction of desires, beliefs, and intentions is not handled explicitly. Semantically, VSK is based on multimodal sorted first order logic (Wooldridge and Lomuscio 2000) and for temporal aspects it includes the possible worlds semantics, i.e., beliefs resp. propositions about knowledge follow weakS5 (KD-45) modal system (Meyer et al. 1991). In spite of the convincing concept of VSK, the underlying multimodal first-order logic suffers from the well-known logical omniscience problem of weakS5 as well as semi-decidability of first order logic.

In the context of autonomous logistics, a key characteristic of the agents is their physical or virtual mobility. Due to the physical movement of logistics objects, a formal approach has to consider ad hoc networks or restricted visibility of the environment. Furthermore, the mobility of agents within the virtual community has to be considered. The VSK model allows for dynamic manipulation of the accessibility of the environment through the visibility function. Petsch introduced an approach for modeling open agent societies with explicit migration in the formal model including representations of real-world organizations on the formal basis of VSK (Petsch 2006).

The internal decision behavior of agents is in focus of the distributed artificial intelligence community. This also reflects the majority of formal approaches which are focused on enabling intelligent behavior within agents (van der Hoek and Wooldridge 2003), (Wooldridge and Jennings 1995), (Rao and Georgeff 1998), (Fisher and Ghidini 2002), (Nide and Takata 2002), and (Timm 2001). Design of intelligent agents is often based on an explicit, cognitive model of beliefs, desires, and intentions, which are based on (Bratman 1987). BDI-agents use a formal semantics and implement a cognitive model of beliefs, desires, and intentions. The underlying idea is that an agent is creating an explicit world model (beliefs) on the basis of observations and its actions. Additionally, it contains a set of objectives (desires or persistent goals) and a set of goals which are currently pursued (intentions). The agent pursues its goals by autonomously created plans. This decision behavior is outlined in Table 2. BDI-agents are “the dominant force” in formal approaches (d’Inverno et al. 2004). Following (Wooldridge 2000) this is caused by their foundation on a widely accepted theory of rational actions of humans, the “great” number of successful complex applications and the availability of a large family of well-understood, sophisticated, and formalized approaches.

```
beliefs := beliefs'
while (true) do
    get next perception p1;
    beliefs := belief-revision-function(beliefs, p1);
    intentions := deliberate(beliefs);
    plan := planning(beliefs, intentions);
    execute(plan);
end while
```

Tab. 2. Simplified decision behavior of BDI agents (Wooldridge 2000)

In 2006, Henesey performed a survey on agent approaches in logistics (Henesey 2006). One of the main conclusions of this survey is that the majority of the agent approaches focus on operation decision support and only rare approaches have been applied in praxis. With respect to the levels of autonomy, the tactical and strategic level as implemented by BDI agents seem to be beneficial to autonomous logistics, as complex internal decision behavior can be modeled explicitly. However, BDI approaches do not focus on system behavior but on agent internal knowledge representation and decision making. In autonomous logistics, there are organizational structures as well as a centralized management defining the boundaries for individual agents’ behavior. Here the BDI approach lacks explicit modeling of utility function or mechanisms for reliable behavior of a group of agents. In (Timm 2004) as well as (Scholz et al. 2006) the formal models of BDI especially the logical framework Lora (Wooldridge 2000) and VSK have been integrated as a unified formal basis for systems of intelligent agents.

In the research of the priority research program on “Intelligent Agents and Business Applications” from 2000 to 2006, it has been stated, that flexibility is the key benefit of intelligent agents (Kirn et al. 2006). However the question arises, if the optimization of individual performance within an agent also leads to a global optimization for a group of agents. In current approaches especially in the context of the CRC on autonomous logistics, we are investigating strategic management in multiagent systems. The strategic management is based on autonomous adjustment of the agent’s autonomy. The underlying model is based on a social mechanism for reflection within social systems and has been transferred to multiagent theory (Timm & Hillebrandt, 2006).

Conclusion

In this chapter, we have discussed two approaches to modelling autonomy in software systems: multiagent systems and communities of autonomous units. The former is a well-known and widely used logical framework in artificial intelligence. The latter is a rule-based and graph-oriented method recently introduced in the context of the Collaborative Research Centre *Autonomous Cooperating Logistic Processes* (cf. Chapter XXX).

As the very first observation in comparison of the two approaches, it has turned out that communities of autonomous units form executable structural models of the axiomatic notion of multiagent systems so that the former provide platform-independent realizations of the latter.

To shed more light on the significance of these observations, future studies will have to work out the relationship in more detail. This will include on one hand to prove that communities of autonomous units do not only follow the structure of multiagent systems, but satisfy also the requested properties. On the other hand, one may employ the well-working decision-making procedures of agents as control mechanism of autonomous units to widen the spectrum of possibilities with respect to the self-control.

References

Agha GA (1986) ACTORS: A Model of Concurrent Computation in Distributed Systems. The MIT Press: Cambridge, Massachusetts

Bratman ME (1987) Intentions, Plans and Practical Reason. Harvard University Press: Cambridge, Massachusetts

Castelfranchi C, Conte R (1992) Emergent functionality among intelligent systems: Cooperation within and without minds. Journal on Artificial Intelligence and Society, 6 (1), pp.78–87

d'Inverno M, Luck M, Georgeff M, Kinny D; Wooldridge M (2004) The dMARS Architecture: A Specification of the Distributed Multi-Agent Reasoning System. Autonomous Agents and Multi-Agent Systems, 9 (1-2), pp. 5-53

Dunne PE; Laurence M; Wooldridge M (2003) Complexity Results for Agent Design Problems. In: Annals of Mathematics, Computing & Teleinformatics, vol. 1, No. 1, pp. 19-36

Ehrig H, Kreowski H-J, Montanari U, Rozenberg G (eds) (1999) Handbook of Graph Grammars and Computing by Graph Transformation, vol 3: Concurrency, Parallelism, and Distribution. World Scientific, Singapore

Fisher M; Ghidini C (2002) The ABC of Rational Agent Modelling. In: Castelfranchi C, Johnson WL (eds) Proceedings of the First International Joint Conference on Autonomous Agents & Multiagent Systems (AAMAS 2002). Bologna, Italy, July 15-19, pp. 849-856

Hölscher K, Kreowski H-J, Kuske S (2006) Autonomous Units and their Semantics – the Sequential Case. In: Corradini, A, Ehrig H, Montanari U, Ribeiro L, Rozenberg G (eds) Proc. 3rd International Conference on Graph Transformations (ICGT 2006), Lecture Notes in Computer Science vol 4178, Springer, Berlin Heidelberg New York, pp 245-259

Janssens D, Kreowski H-J, Rozenberg G (2005) Main Concepts of Networks of Transformation Units with Interlinking Semantics. In: Kreowski H-J, Montanari U, Orejas F, Rozenberg G, Taentzer G (eds) Formal Methods in Software and System Modeling, Lecture Notes in Computer Science vol 3393. Springer, Berlin Heidelberg New York, pp 325-342

Knirsch P.; Timm IJ (1999) Adaptive Multiagent Systems Applied on Temporal Logsitics Networks. In: Muffato M, Pawar KS (eds) Logistics in the Information Age. SGE Ditoriali: Padova Italy, pp. 213–218

Kuske S (2000) Transformation Units-A Structuring Principle for Graph Transformation Systems. Ph.D. thesis, Bremen

Maes P (1994) Agent that Reduce Work and Information Overload. In: Communications of the ACM 37, Nr.7, pp. 31–40

Meyer J-JC, van der Hoek W, Vreeswijk GAW (1991) Epistemic Logic for Computer Science: A Tutorial (Part One). In: Bulletin of the EATCS 44. European Association for Theoretical Computer Science, pp. 242–270

Müller JP (1996) The Design of Intelligent Agents: A Layered Approach. Lecture Notes in Artificial Intelligence 1177. Springer: Berlin

Müller H-J (1997) Towards agent systems engineering. Data & Knowledge Engineering, 23(3):217–245

Nide N, Takata S (2002) Deduction Systems for BDI Logics Using Sequent Calculus. In: Castelfranchi C, Johnson WL (eds) Proceedings of the First International Joint Conference on Autonomous Agents & Multiagent Systems (AAMAS 2002). Bologna, Italy, July 15-19, pp. 928-935

Patel-Schneider PF, Hayes P, Horrocks I (2007) OWL Web Ontology Language Semantics and Abstract Syntax. W3C-Recommendation. http://www.w3.org/TR/owl-semantics/ (last visited Jan. 2007)

Petsch M (2006) Integrative Betrachtung der Offenheit von Multiagentensystemen unter technischen, systemischen, sozialen, und organisatorischen Aspekten. Dissertation. Technische Universität Ilmenau, Fachbereich Wirtschaftswissenschaften, Institut für Wirtschaftsinformatik, September

Rao A, Georgeff M (1998) Decision procedures for BDI logics. In: Journal of Logic and Computation 8, No. 3, pp. 293–342

Rovatsos M, Weiss G (2005) Autonomous Software. In Chang S K (ed): Handbook of Software Engineering and Knowledge Engineering, volume 3: Recent Advances. World Scientific Publishing: River Edge, New Jersey

Russel SJ, Norvig P (1994) Artificial Intelligence: A Modern Approach. Prentice Hall: Englewood Cliffs

Russel SJ, Norvig, P (2003) Artificial Intelligence: A Modern Approach. 2nd Edition. Prentice Hall: Englewood Cliffs, New Jersey

Singh, MP, Rao AS, Georgeff M P (1999) Formal Methods in DAI: Logic-Based Representation and Reasoning. In: Weiss G (ed) Multiagent Systems - A Modern Approach to Distributed Artificial Intelligence. The MIT Press: Cambridge, Massachusetts

Smith RG (1988) The contract net protocol: High-level communication and control in a distributed problem solver. In Bond A H, Gasser L (eds.) Readings in Distributed Artificial Intelligence. Morgan Kaufmann: San Mateo

Smith RG, Davis R (1980) Frameworks or cooperation in distributed problem solving. IEEE Transactions on Systems, Man and Cybernetics, 11(1)

Timm IJ (2001): Enterprise Agents Solving Problems: The cobac-Approach. In: Bauknecht K, Brauer W, Mueck Th (eds): Informatik 2001 - Tagungsband der GI/OCG Jahrestagung, 25.-28. September 20011, Universität Wien, pp. 952-958

Timm IJ (2004) Dynamisches Konfliktmanagement als Verhaltenssteuerung Intelligenter Agenten. DISKI 283, infix: Köln

Timm IJ (2006) Strategic Management of Autonomous Software Systems. Technical Report. TZI Bericht Nr. 35, Universität Bremen, Bremen

Timm IJ, Hillebrandt F (2006) Reflexion als sozialer Mechanismus zum strategischen Management autonomer Softwaresysteme. In: Schmitt M., Michael F, Hillebrandt F (eds) Reflexive soziale Mechanismen. Von soziologischen Erklärungen zu sozionischen Modellen, VS Verlag: Wiesbaden, pp.255-288

Timm IJ, Scholz T, Herzog O (2006) Capability-Based Emerging Organization of Autonomous Agents for Flexible Production Control. In: Advanced Engineering Informatics Journal, 20, S. 247-259

van der Hoek W, Wooldridge M (2003) Towards a Logic of Rational Agency. In: International Logic Journal of the IGPL, 11, Nr. 2, pp. 133-157

Wooldridge MJ (2000) Reasoning about Rational Agents. The MIT Press: Cambridge, Massachusetts

Wooldridge M, Jennings N (1995) Intelligent Agents: Theory and Practice. In: The Knowledge Engineering Review 10, No. 2, pp. 115–152

Wooldridge M, Lomuscio, A (2000) A Logic of Visibility, Perception, and Knowledge: Completeness and Correspondence Results. In: Proceedings of the Third International Conference on Pure and Applied Practical Reasoning (FAPR-2000). London, UK