I am autonomous, you are autonomous

Hans Weigand[1], Virginia Dignum[2],

[1] Infolab, Tilburg University, PO Box 90153,
5000 LE Tilburg, The Netherlands
h.weigand@uvt.nl
[2] University Utrecht, Intelligent Systems Group, PO Box 80089,
3508 TB Utrecht, The Netherlands
Virginia@cs.uu.nl

Abstract Autonomy is regarded as a crucial notion in multi-agent systems and several researchers have tried to identify what are the agent's parts that give it an autonomous character. In this paper, we take a different approach. If we assume that agents are autonomous (and this is a quite reasonable assumption in many practical situations, such as e-commerce), the more interesting question is: how to cope with the autonomy of agents? What are the effects on the way agents have to coordinate their behavior with other agents, and on the agent design process? And what are the effects of that (secondary effects) on the architecture of agents and agent societies. We address these questions by working out the concept of „collaboration autonomy", and by describing an agent society model that respects this kind of autonomy.

1 Introduction

Why is autonomy still a poorly understood concept? Perhaps because autonomy is such a loaded term in Western culture: individual autonomy was the dream of the Enlightenment, and it is still a cornerstone of most of the Western political and social theories, although it is not a univocal concept at all. It might be argued (but not here) that our problems in understanding agent autonomy are symptomatic of the dilemmas that we face daily in dealing with human autonomy in our societies.

In his seminal article on agent autonomy [4], Castelfranchi separated autonomy from agenthood. In his view, not all agents are autonomous. So he asked himself the question what makes an agent autonomous. In his view, the most interesting kind of autonomy is goal-autonomy, which itself is dependent on the „non-negotiability of beliefs". This means „we cannot believe a certain assertion just because there would be some advantage for us in believing it". Unfortunately, Castelfranchi does not motivate this axiom, and he also has no argument for why this property that he states about humans would also hold for agents. It seems that human autonomy at some point cannot be further analyzed.

The contribution of this paper is twofold. First, we present our position on the agent autonomy issue, and secondly, we provide a short overview of the OperA model for agent societies and how it meets autonomy requirements.

2 Towards a transactional analysis of autonomy

Traditionally, agent autonomy is viewed as a *property* of some piece of software, that is one of the features that the software must fulfil to be considered an agent. This descriptive property can also be interpreted as a design requirement: you can build an agent in any way you want, but it must exhibit the autonomy property in the end. In this section, we make a few remarks about this software property approach, and then propose another way of looking at autonomy, inspired by the psychological approach of transactional analysis.

2.1 Autonomy as self-governance

Abdelkader [1] discusses two interpretations of autonomy: *self-governance* (the agent is steered in selecting what goals have to be achieved by a set of motivations), and *independence* (the agent is independent from other agents) – and he prefers the first one, as independence is not a sufficient criteria. Carabelea et al [3] identify five forms of autonomy: user-autonomy, social autonomy, norm-autonomy, self-autonomy and environment autonomy. They rightly state that autonomy is a relational property (in line with [4]): X is autonomous *from* (other agent or object) Y *for* p (the object of autonomy), and they define this further as: X's behaviour regarding p cannot be imposed by Y. They call this an external perspective, which should be supplemented by an internal perspective. They suggest that one should be able to identify what are the agent's parts that give it an autonomous character (e.g. a goal maintenance module).

We agree that autonomy is not the same as independence. If autonomy is interpreted as *independence*, then the whole enterprise of agent societies is doomed to failure. After all, what makes a society interesting, for its members or for others, is that there are certain dependencies [4]. What is often overlooked in the discussions, is that these dependencies work to ways: not only the agent is dependent on its environment, but also the environment (other agents) will depend on the agent., to some extent. In other words, the agent will perform a certain role, a function with an added-value.

A controversial assumption is that the agent autonomy should be reflected in its architecture. One could argue against this assumption that if autonomy means anything, the agent is „master in his own house", so it seems odd to pose any requirement on the internal architecture. Of course, it is nice if a goal maintenance module, obligations, norms etc can be identified in the architecture, but the fact that these modules can be identified does not prove anything yet about autonomy, because it does not tell how the agent deals with them. If the „should" is taken as a design principle – if the intended meaning is that agent designers can better use such an

architecture – then the assumption may be valid, but it must be clear that this is in no way a guarantee of autonomy nor an absolute requirement.

It should be noted that the notion of self-governance (behavior cannot be imposed by another party) assumes an intensional stance. Let s be a stimulus from another object (we will restrict ourselves to other agents) that aims at behavior x (for example, a REQUEST(x) message). The software receiving such a message will decide somehow whether to perform x or not (we may assume that it is capable of performing x). In traditional systems, this typically depends on authorizations and user roles, but the decision procedure can be made much more subtle, for example, by including a computation of the utility of doing x. If no stimulus has effect, we again have some „independence" situation that is not very interesting, so we can assume that at least some stimuli do have effect. In that case, the other agent can choose exacty these stimuli, in that way impose the required behavior, and there seems be no room for autonomy. So even if the agent performs only behavior congruent with its goals or utility function, and one may think that therefore it behaves autonomously, it can still be manipulated, and to say whether the behavior is self-chosen or imposed is all in the eyes of the beholder. Therefore, we think that also the criterion of self-governance is not useful as a litmus test for autonomy. However, the question is whether such a litmus test is what we need.

2.2 A new approach

Rather than *assuming* autonomy to be a *required* property of agents, and from there infer some architecture that would *guarantee* this property, we propose a radically different approach: namely, to *require* autonomy as an *assumed* property of agents, and to infer from there some architecture that *respects* this property.

Why would we assume autonomy?
Because software programs are deployed by human users and human users are deemed to be autonomous. In most of the practical applications of agents, e.g. in e-commerce, the agent is an agent *of* some human user. For example, the agent has to buy an item on the Internet, or has to sell it in virtual shop. The same is true for a robot in a production plant or on the battle field. If the software we meet (or our software meets) is the agent of another human, then we cannot assume that it simply performs every request we make, and that it will always live up to its commitments. Hence we better not assume that to be the case, in other words, *assume* its autonomy. We don't have to make this assumption in a simulated world of programs that we control ourselves. But as soon as we enter the real world, we are dealing with software belonging to other humans or human organizations.

Why would we require *our agents to assume autonomy?*
We argued that it is a reasonable assumption that the piece of software you encounter on the Internet or anywhere in the real world behaves autonomously, but what if you do not? Is it not possible to manipulate this software, as it was sketched above? Yes, we agree that it is possible, and even if the human owner of the sofware is closely monitoring and controlling its behavior, one might attempt to manipulate software

and owner together. But the question is not whether it is possible, but whether it is desirable. There are two important reasons why it is not. First, manipulating the agent quickly becomes manipulating the human owner, and this violates the ethical principle that humans should always be treated as a subject, not as an means (Kant). Secondly, you can at most try to manipulate, but you are not sure that you will succeed. Therefore, it is also safer to assume that the piece of software is autonomous, because you may very well miss your own target if you do not make that assumption. Let us get more practical: if you deploy a shopping agent, then you better build it in such a way that it does not naively assume that when something is advertized in a web shop, it is also available. Or that when the shop has promised to deliver the item, this will always happen. Instead, it is better to design your agent in such a way that it can cope with contingencies caused by the other agents autonomous decisions. That is, to ask first whether the item is available. To ask for a firm commitment in the case of a risky transaction. To look for trust-enablers. Trust is only relevant in a situation of potential risk, so by using trust-enabling mechanisms you already assume a certain autonomy of the other party. This second argument (that it is safer) is not only valid in the case of cooperative agents, but also if you want them to be opportunistic and adversarial – even then, you don't want your agent to be naive about its opponents. As it is simply *better* to assume autonomy, we require our agents to do so. This holds for agents operating within some society (and their designers), but also for agent governing an agent society (and its designer).

How do we respect agent autonomy?
By turning around the question and requesting our agents to assume the autonomy of other agents, the next question is: what are the consequences for the agent architecture? The question is not the old one: what requirements do we put on a certain piece of software before we call it an autonomous agent? Rather the question becomes: how do we cope with the autonomy of agents? Or, in other words, how do we build agents that respect (other) agent autonomy?

In section 3, we will try to answer these questions in the context of OperA, a framework for building agent societies. The requirements that we formulate there do not have the goal of *guaranteeing* autonomy but rather of *coping with* the given autonomy. In our view, agent contracts become highly relevant then. Similarly, we can imagine requirements on the individual agents within a society that deal with other agents. To respect autonomy means at least to be prepared that any request (or assertion) *can* be rejected. From there it follows that you must also be prepared that your request can be accepted. Hence your agent must support a protocol that includes accepts and rejects. This can be done at different levels of sophistication.

2.3 I am autonomous, you are autonomous

"I'm OK, you are OK" is a popular expression originating from the Eric Berne's theory of transactional analysis [2]. Berne abandoned psychoanalytic theory in favor of a theory centered on communication. He focused on the information that is exchanged between people and conceptualized and categorized it in terms of transactions. By isolating transactional stimuli and responses he provided us with a

method with which to study how people influence each other, and made possible the fine-grained analysis of person-to-person communication. People often adopt socially dysfunctional behavioral patterns called "games". For Berne, *individualism* means that a person is acting within a dysfunctional life script, maintaining a belief that others are not OK. In contrast, *autonomy* means in this theory that the individual is in tune with himself, others and the environment and therefore is acting freely (script free).

We refer to transactional analysis not because we think it is directly applicable to the agent autonomy question, but because of the interesting paradigm shift it offers. Autonomy is not viewed as an individual property, not even a relationship, but as a something that governs interactions. We just argued that autonomy is not something that must be required from agents, but that must be assumed. "You are autonomous" – that means that I will treat you as a subject, which does not follow my requests slavishly, but only may do so when I am able to provide good reasons. "I am autonomous" – that means also that I will not slavishly obey your requests, it means that I am able to enter meaningful relationships – call it contracts, which I expect you to respect. In other words, what we propose is that agent autonomy is first of all viewed as a norm governing agent interactions in an agent society. Agents don't need to prove that they are autonomous, but they (including their designers and agent society designers!) should live up according to this norm. This has certain consequences for the architecture of the agent society. It also imposes a norm on that piece of software representing an individual agent, although it does not specify how its designer is going to fulfill this norm.

Transactional analysis can help to see the goal autonomy in perspective: the agent has its own goals (as it is a piece of software, these goals are ultimately delegated to it by a human actor), but it also has to perform a role, deliver a function within the society it finds itself. This function is to the benefit of other agents. How can this paradox be solved? Basically, there are two choices: either the agents goals are completely congruent with the function it offers, leading to so-called benevolent agents that can hardly be called autonomous – or the paradox is solved by organizing agent transactions as *exchanges*: because in the exchange both the goal of the agent himself and the benefit of the other agent can be equally respected. The institutional way of organizing exchanges is by means of contracts, and contracts play a central role in our agent society model.

3 Autonomy in OperA

In this section, we briefly describe the model for agent societies **OperA** (**O**rganizations **per** **A**gents) [8].[1] This framework emerges from the realization that in organizations interactions occur not just by accident but aim at achieving some

[1] The name illustrates the dual relation between organizations and agents, the fact that organizations are outmost dependent on its agents, but, as in a musical opera, a script is needed that guides and constrains the performance of the actors, according to the motivations and requirements of the society designer.

desired global goals, and that participants are autonomous, heterogeneous and not under the control of a single authority.

The purpose of any society is to allow its members to coexist in a shared environment and pursue their respective goals in the presence or in co-operation with others. A collection of agents interacting with each other for some purpose and/or inhabiting a specific locality can be regarded as a society. Societies usually specify mechanisms of social order in terms of common norms and rules that members are expected to adhere to [5]. An organization can be defined as a specific solution created by more or less autonomous actors to achieve common objectives. Organizational structure can therefore be viewed as a means to manage complex dynamics in (human) societies. This implies that approaches to organizational modeling must incorporate both the structural and the dynamic aspects of such a society.

From an organizational perspective, the main function of an individual agent is the enactment of a role that contributes to the global aims of the society. That is, society goals determine agent roles and interaction norms. Agents are actors that perform role(s) described by the society design. The agent's own capabilities and aims determine the specific way an agent enacts its role(s), and the behavior of individual agents is motivated from their own goals and capabilities, that is, agents bring in their own ways into the society as well. However, a society is often not concerned about which individual agent will actually play a specific role as long as it gets performed. Several authors have advocated role-oriented approaches to agent society development, especially when it is manifest to take an organizational view on the application scenario [6] [11].

The above considerations can be summarized in the recognition that there is a clear need for multi-agent frameworks that combine and use the potential of a group of agents for the realization of the objectives of the whole, without ignoring the individual aims and 'personalities' of the autonomous participant agents. That is, in order to represent interactions between agents in such an open context, a framework is needed that meets the following requirements:

- **internal autonomy requirement**: interaction and structure of the society must be represented independently from the internal design of the agents
- **collaboration autonomy requirement**: activity and interaction in the society must be specified without completely fixing in advance the interaction structures.

The first requirement relates to the fact that since, in theory, an open society allows the participation of multiple, diverse and heterogeneous entities, the number, characteristics and architecture of which are unknown to the society designer, the design of the society cannot be dependent on their design. With respect to the second requirement, fundamentally, a tension exists between the goals of the society designer and the autonomy of the participating entities. On the one hand, the more detail the society designer can use to specify the interactions, the more requirements are possible to check and guarantee at design time. This allows, for example, to ensure the legitimacy of the interactions, or that certain rules are always followed [10]. On the other hand, there are good reasons to allow the agents some degree of freedom, basically to enable their freedom to choose their own way of achieving collaboration, and as such increase flexibility and adaptability.

The OperA approach consists of a 3-layered model that separates the concerns of the organization from those of the individual. The top layer, called the Organizational Model, describes the structure and objectives of a system as envisioned by the organization, and the bottom layer, the Interaction Model, the activity of the system as realized by the individual agents. In order to connect individual activity with organizational structure we add a middle layer, the Social Model that describes the agreed agent interpretation of the organizational design. Fig. 1 depicts the interrelation between the different models. In the following subsections, we will describe each of the three models in more detail.

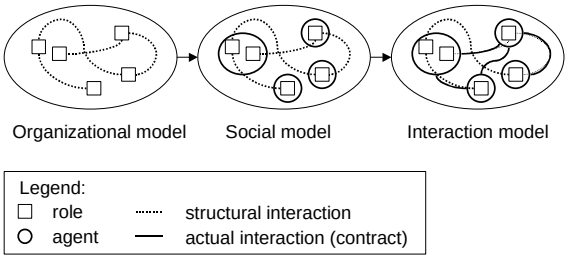

Fig. 1. Organizational framework for agent societies

An OperA model can be thought of as a kind of abstract protocol that governs how member agents should act according to social requirements. Interaction is specified in contracts, which can be translated into formal expressions (using the logic for contract representation, described in [7]), and therefore ensure that compliance can be verified.

The actual behavior of the society emerges from the goal-pursuing behavior of the individual agents within the constraints set by the Organizational Model. From the society point of view, this creates a need to check conformance of the actual behavior to the desired behavior, which has several consequences. Firstly, we have to make explicit the commitments between agents and the society. An actor is an agent performing one or more roles in the society. The objectives and capabilities of the role are described in the OM and the actual agreements of the actor concerning its interpretation of the role are explicitly described in a social contract. We use the term Social Model to refer to the social contracts that hold at a given moment in a society. Secondly, actual interaction between actors must also be made explicit, which can be done through (bilateral) contracts as well. We call this the Interaction Model (IM). Checking the conformance of actual behavior to the desired behavior can now be realized in two steps:

- Checking whether the contract set up by two or more agents conforms to the OM. The OM can be more or less elaborate, depending on the type of society. Typically, it does not provide more than a few "landmarks" that describe the main features (conditions, obligations, and possibly partial plans) of interaction between roles.
- Monitoring whether the actual messaging behavior conforms to the contract. This is primarily a responsibility of the agents themselves. Depending on the type of society, the OM can constraint the monitoring or even assign it to the society.

3.1 The Organizational Model

Starting point of the Agent Society Model is the organizational model (OM) that describes the structure and global characteristics of a domain from an organizational perspective. That is, from the premise that it is the society goals that determine agent roles and interaction norms. The organizational model is based on the analysis of the domain in terms of the coordination and normative elements. The OM specifies the global objectives of the society and the means to achieve those objectives.

The OM specifies an agent society in terms of four structures: social, interaction, normative and communicative. The social structure specifies objectives of the society, its roles and the model that governs coordination. The global objectives of an organization are represented in terms of objectives of the roles that compose the organization. Roles are tightly coupled to norms, and roles interact with other roles according to interaction scripts that describe a "unit" of activity in terms of landmarks. The interaction structure gives a partial ordering of the scene scripts that specify the intended interactions between roles. Society norms and regulations are specified in the normative structure, expressed in terms of role and interaction norms. Finally, the communicative structure specifies the ontologies for description of domain concepts and communication illocutions. The way interaction occurs in a society depends on the aims and characteristics of the application, determines the relations between roles, and how role goals and norms are 'passed' between related roles. For example, in a hierarchical society, goals of a parent role are shared with its children by delegation, while in a market society, different participants bid to the realization of a goal of another role.

3.2 The Social Model

We assume that individual agents are designed independently from the society to model the goals and capabilities of a given entity. In order to realize their own goals, individual agents will join the society as enactors of role(s) described in the organizational model. This means that several populations are possible for each organizational model. Agent populations of the organizational model are described in the social model (SM) in terms of commitments regulating the enactment of roles by individual agents. In the framework, agents are seen as autonomous communicative entities that will perform the society role(s) according to their own internal aims and architecture. Because the society designer does not control agent design and behavior, the actual behavior of the society instance might differ from the intended behavior. The only means the society designer has for enforcing the intended behavior is by norms, rules and sanctions. That is, when an agent applies, and is accepted, for a role, it will commit itself to the realization of the role goals and it will function within the society according to the constraints applicable to its role(s). These commitments are specified as social contracts that can be compared to labor contracts between employees and companies. The society can sanction undesirable (wrong) behavior as a means to control how an agent will do its 'job'.

The Social Model is defined by the **role enacting agents** (**reas**) that compose the society. For each agent, the rea reflects the agent's own requirements and conditions

concerning its participation in the society. Depending on the complexity of the implemented agents, the negotiation of such agreements can be more or less free. However, making these agreements explicit and formal, allows the verification of whether the animated society behaves according to the design specified in the OM. The SM specifies a population of agents in a society, which can be seen as an instantiation of the OM. When all roles specified in the OM are instantiated to agents in the SM, we say that the SM provides a full instantiation of the society; otherwise, it is a partial instantiation.

3.3 The Interaction Model

Finally, interaction between agents populating a society is described in the interaction model (IM) by means of interaction contracts. This model accounts for the actual (emergent) behavior of the society at a given moment. Interaction agreements between agents are described in interaction contracts. Usually interaction contracts will 'follow' the intended interaction possibilities specified in the organizational model. However, because of the autonomous behaviour of agents, the interaction model must be able to accommodate other interaction contracts describing new, emergent, interaction paths, to the extent allowed by the organizational and social models.

OperA provides two levels of specification for interactions. The OM provides a script for interaction scenes according to the organizational aims and requirements and the IM, realized in the form of contracts, provides the interaction scenes such as agreed upon by the agents. It is the responsibility of the agents to ensure that their actual behaviour is in accordance with the contracts (e.g. using a monitoring agent or notary services provided by the society for that). However, it is the responsibility of the society, possibly represented by some of its institutional roles, to check that the agents fulfill these responsibilities.

The architecture of IM consists of a set of instances of scene scripts (called scenes), described by the interaction contracts between the role enacting agents for the roles in the scene script. An interaction scene results from the instantiation of a scene script, described in the OM, to the reas actually enacting it and might include specializations or restrictions of the script to the requirements of the reas.

4 Conclusion

The OperA model integrates a top-down specification of society objectives and global structure, with a dynamic fulfilment of roles and interactions by participants. The model separates the description of the structure and global behaviour of the domain from the specification of the individual entities that populate the domain. This separation provides several advantages to our framework above traditional MAS models. On the one hand, coordination and interaction in MAS are usually described in the context of the actions and mental states of individual agents [9]. In open societies, however, such approach is not possible because agents are developed independently from the society and there is therefore no knowledge about the internal

architecture of agents, nor possibilities to directly control or guide it. Furthermore, conceptual modeling of agent societies (based on the social interactions) requires that interaction between agents be described at a higher, more abstract level, that is, in terms of roles and institutional rules. On the other hand, society models designed from an organizational perspective reflect the desired behaviour of an agent society, as determined by the society 'owners'. Once 'real' agents populate the society, their own goals and behaviour will affect the overall society behaviour, that is, such social order as envisioned by the society designer is in reality a conceptual, fictive behaviour. From an organizational perspective, the main function of individual agents is the enactment of roles that contribute to the global aims of the society. That is, society goals determine agent roles and interaction norms. Agents are actors that perform role(s) described by the society design. The agent's own capabilities and aims determine the specific way an agent enacts its role(s).

The OperA model can be viewed as an attempt to design agent societies that respect agent autonomy. We do not claim that this is achieved exhaustively in its present form. But it illustrates what a transactional analysis of autonomy could mean in practice.

References

1. Abdelkader, G.: Requirements for achieving software agents autonomy and defining their responsibility. *Proc. Autonomy Workshop at AAMAS 2003*, Melbourne, 2003.
2. Berne, Eric: *Games People Play*. New York, Grove Press, 1964.
3. Carabelea, C. O. Boissier, A. Florea: Autonomy in multi-agent systems: a classification attempt. *Proc. Autonomy Workshop at AAMAS 2003*, Melbourne, 2003.
4. Castelfranchi, C.: Guarantees for Autonomy in Cognitive Agent Architecture. *Proc. ATAL'94*. LNAI 890, Springer 1995.
5. Davidsson, P.: Emergent Societies of Information Agents. Klusch, M, Kerschberg, L. (Eds.): *Cooperative Information Agents IV*, LNAI 1860, Springer, 2000, pp. 143–153.
6. Dignum, V., Meyer, J-J., Weigand, H.: Towards an Organizational Model for Agent Societies Using Contracts. In: *Proc. of AAMAS, the 1st International Joint Conference in Autonomous Agents and Multi-Agent Systems*, Bologna, Italy, 2002.
7. Dignum, V., Meyer, J.-J., Dignum, F., Weigand, H.: Formal Specification of Interaction in Agent Societies. In: Hinchey, M., Rash, J., Truszkowski, W., Rouff, C., Gordon-Spears, D., (Eds.): *Formal Approaches to Agent-Based Systems (FAABS'02)*. LNAI 2699, Springer-Verlag, 2003.
8. Dignum, V.: *A Model for Organizational Interaction: Based on Agents, Founded in Logic*. PhD thesis, Utrecht University, 2004.
9. Ferber, J., Gutknecht, O.: A meta-model for the analysis and design of organizations in multi-agent systems. *Proc. of ICMAS'98*, IEEE Press. 1998.
10. Weigand, H., Dignum, V., Meyer, J-J., Dignum, F.: Specification by Refinement and Agreement: Designing Agent Interaction Using Landmarks and Contracts. In: Petta, P., Tolksdorf, R., Zambonelli, F. (Eds.): *Engineering Societies in the Agents World III: Proceedings ESAW'02*, LNAI 2577, Springer-Verlag, 2003, pp. 257-269.
11. Zambonelli F., Jennings, N., Wooldridge, M.: Organisational Abstractions for the Analysis and Design of Multi-Agent Systems. In: Ciancarini P., Wooldridge, M. (eds.): *Agent-Oriented Software Engineering*, LNCS 1957, Springer-Verlag, 2001, pp. 235 – 251.