## Towards an architecture for self-regulating agents: a case study in international trade

Burgemeestre, C.B.; Hulstijn, J.

***published in***
Lecture Notes in Computer Science
2009

***document version***
Publisher's PDF, also known as Version of record

# Towards an Architecture for Self-regulating Agents: A Case Study in International Trade

Brigitte Burgemeestre[1], Joris Hulstijn[1], and Yao-Hua Tan[1,2]

[1] PGS IT Audit, Faculty of Economics and Business Administration, Vrije Universiteit
[2] Department of Technology, Policy and Management, Delft University of Technology
{jhulstijn,cburgemeestre}@feweb.vu.nl, Y.Tan@tudelft.nl

**Abstract.** Norm-enforcement models applied in human societies may serve as an inspiration for the design of multi-agent systems. Models for norm-enforcement in multi-agent systems often focus either on the intra- or inter-agent level. We propose a combined approach to identify objectives for an architecture for self-regulating agents. In this paper we assess how changes on the inter-agent level affect the intra-agent level and how a generic BDI architecture IRMA can be adapted for self-regulation. The approach is validated with a case study of AEO certification, a European wide customs initiative to secure the supply chain while facilitating international trade.

**Keywords:** Self-regulation, agent architectures, compliance.

## 1 Introduction

To make autonomous agents comply with norms, various enforcement mechanisms have been proposed. Norms here are defined as standards of behavior that are acceptable in a society, indicating desirable behaviors that should be carried out, as well as undesirable behaviors that should be avoided [14]. Enforcement mechanisms often require special 'observers' or 'regulator agents' that actively monitor the behavior of the other agents, and sanction them in case of norm violations. When developing norm enforcement mechanisms for multi-agent systems, the modeling is often focused on the inter-agent level (between agents). Models aim to construct norm enforcement mechanisms by agent interaction. The intra-level (inside the agent) is mainly treated as a black box. We argue that the intra- and inter-agent aspects cannot be viewed separately, especially in norm enforcement where external stimuli should motivate an agent to adapt its behavior and thereby its internal mechanisms.

Norm-enforcement models applied in human societies may serve as an inspiration for the design of electronic institutions and open agent systems. An enforcement mechanism that is based on an agent's internal architecture to achieve compliant behavior, and does not require additional 'observers' is self-regulation. Self-regulation is a control approach in which rule making and enforcement are carried out by the agent itself, instead of by the regulator. Self-regulation is an alternative for direct control, when external supervision and enforcement is not possible, ineffective or when there is a lack of controlling resources. For example, in an e-institution it might be impossible to check all agent actions for compliance in real time. A solution might

J. Padget et al. (Eds.): COIN 2009, LNAI 6069, pp. 320–333, 2010.

be to do a code review and determine whether an agent is compliant by design [17]. In human societies, programs of self–regulation have been found to contribute to expanded control coverage and greater inspectorial depth [3]. Self-regulation can be implemented in various ways, ranging from voluntary self-regulation, where a group of agents chooses to regulate themselves, to mandated or enforced self-regulation, where a government agency delegates some of its regulative and enforcing tasks to the agents subjected to the norm, but retains the supervision [16]. Each model of self-regulation causes different dependencies among agents and different information needs, which imposes different requirements on the agent architecture.

A specific case of self-regulation is the Authorized Economic Operator (AEO) program [12]. The AEO program is a European wide customs initiative that aims to secure international trade while at the same time reducing the administrative burden for companies through the use of self-control. Companies that are trustworthy in the context of customs related operations and have a good internal control system may apply for the AEO certificate and receive operational benefits from simplified customs procedures, preferential treatment, and less physical inspections. Companies that do not have an AEO certificate remain subject to the current level of customs controls. Participation in the AEO program is voluntary, but demonstrating effective self-control mechanisms is a necessary requirement.

Implementing self-regulation as a control mechanism thus results in a redistribution or delegation of control tasks among the actors. Agents have to adapt their internal mechanisms to cope with these tasks. We see that changes at the inter-agent level affect the intra-level. We therefore propose to use a combined approach to develop an architecture which can use self-regulation as a control mechanism for multi-agent systems. The research questions we would like to answer are:

1. What objectives need to be met by an architecture for self-regulating agents?
2. How should the existing BDI-agent architecture be adapted for self-regulation?

We use a combination of frameworks to cover the inter- as well as intra-agent aspects. For intra-agent analysis the Intelligent Resource-Bounded Machine Architecture (IRMA) [4] is a good starting point, because it is a general BDI architecture [15], which is well accepted and forms the basis for more recent agent architectures, such as AgentSpeak or 3APL. Software engineering methodology TROPOS [5] provides suitable concepts to analyze agents' dependencies at the inter-agent level.

The remainder of the paper is structured as follows. In Section 2 we analyze the difference between direct control and self-regulation using TROPOS. Using this analysis we generate objectives for the internal architecture of a self-regulating agent. We apply these objectives to IRMA and propose some adaptations (Section 3). Using the extended architecture and the TROPOS model, we analyze a case study of AEO (Section 4). We examine if our adapted version of the architecture covers the findings of the case study. We identify its suitability and shortcomings.

## 2 Inter-agent Analysis

We first analyze the types of agents involved in regulation, and the dependencies between them. To do so we use concepts from the early requirements phase of the TROPOS methodology [5], which is derived from the *i** framework [18]. The key

concepts we use are: actor, goal, plan, resource and dependency. An actor can be an autonomous agent that has a goal or strategic interest, based on its organizational role. A goal can be satisfied through the execution of a plan, which is an abstract representation of a way of doing something. A resource can be a physical or informational entity. Actors can depend on each other to reach a certain goal, to execute a plan or to obtain resources. The agent that depends on another agent is called the depender, the agent he depends on is called the dependee. The object which is the subject of the dependency relation is called the dependum.

We first model direct control, where actions of autonomous agents are regulated by special regulator agents. After that we analyze self-regulation and assess what should change when an autonomous agent internalizes control tasks of the regulator agent.

### 2.1 Agents' Dependencies in Direct Control

In direct control we have two types of agents: an Actor agent (A) that is carrying out an activity and a Regulator agent (R) that is responsible for regulating A's actions such that agent A complies with the norms that are applicable to A. An agent can violate the norms through pursuing an illegal goal or by performing an illegitimate action. We assume that R has a norm framework from which it derives the set of norms tailored to an agent's specific situation. To regulate A, agent R must have plans for executing the following activities: R1 'Specify norms for actor', R2 'Determine control indicators of actor', R3 'Monitor actor's actions' and R4 'Sanction actor'. R1 generates a set of norms for A. R uses information about A and A's actions to select the appropriate norms from the norm framework that apply to A's specific situation. R2 determines appropriate 'control indicators'. A control indicator is the kind of evidence required to demonstrate compliance of a norm, as well as infrastructural requirements to collect that evidence. For example: when a company sends an invoice, they make a copy of the invoice and store the copy to be able to check later whether all collected payments are correct and complete. R3 concerns the monitoring performed by R on A's actions, based on information provided by A about the control indicators. R4 describes the plan of R to sanction A in case of a norm violation. Agent A's model is quite simple, as A is a 'blind' agent that has no knowledge about the norms or control indicators and can only act. Therefore it is possible that A unknowingly engages in an activity that violates a norm that is imposed upon A by R. However, we do assume that A remembers action-sanction relations and that it can decide to cancel an action that will lead to a sanction.

Figure 1 shows the results of the dependency analysis for direct control.

A consequence of this division of tasks, where the regulator is responsible for the majority of control tasks, is that it is very labor intensive for the regulator. The regulator needs to specify norm sets and control indicators for all agents and needs to do all the verification and auditing. Furthermore the suitability of the rules can be disputable. Overregulation can occur when all the characteristics of the individual agents and possible exceptions need to be taken into account in the rules. Or rules can become ill fitting when they are supposed to be suitable for the majority of the agents but turn out to be compromises, which are not suitable for any individual agent. In direct control relations, Actor agents often have little influence on the rules that are assigned to them and simply have to adapt to the rules that are given.

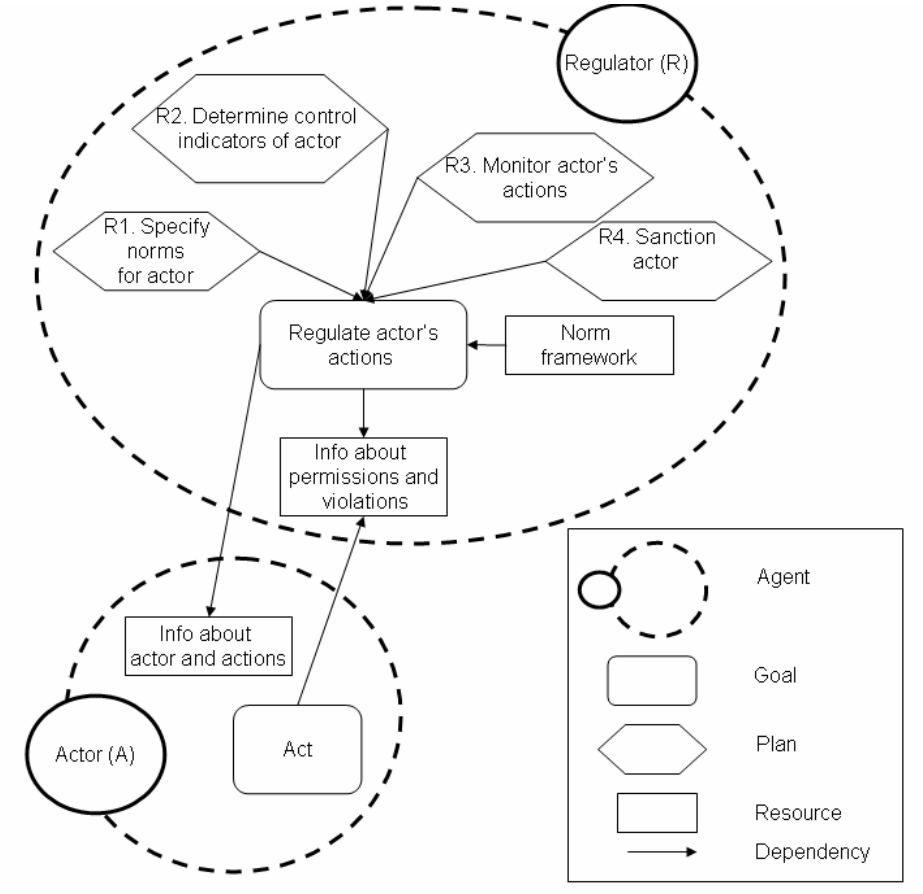


**Fig. 1.** TROPOS model of direct control. The actions of an actor (A) are regulated by a regulator (R). Note that arrows depict dependency, not information flow. So to regulate A's actions, R depends on A for information about actor and actions.

### 2.2 Agents' Dependencies in Self-regulation

For self-regulation we start again with two types of agents: the actor agent (A) and the regulator agent (R). In self-regulation several control tasks are delegated from R to A. Since A is autonomous, R can never be absolutely certain that A complies. R thus has to implement a mechanism to make A regulate itself appropriately. Furthermore, to maintain the power of the regulator to handle non-compliant agents, the sanctioning task (R4) remains the regulator's responsibility.

We first consider the consequences of the internalization of control tasks by A. Plans R1, R2 and R3 may be internalized by agent A as plans: A1 'Specify norms', A2 'Determine control indicators' and A3 'Monitor actions'. A1 specifies norms based on a norm framework which originates from R. This entails a new dependency between A and R: A now depends on R for communicating the norm framework.

When the norm specification is done by A, A is also supposed to be able to differentiate between norm violations and norm compliance. A therefore no longer depends for information about violations and permissions on R, but has to do it himself. A2 defines control indicators about A's actions, based on the norms defined in A1. A3 describes the monitoring actions of A which it performs in the context of the control indicators from plan A2. The plans A1, A2, and A3 together, should support A to act in compliance with the norms. The acts of A in turn affect the nature of the control actions. If A starts doing different activities the control indicators may become less effective and A therefore has to determine new control indicators that cover the norms. For example, if A replaces the process of sending paper invoices to its customers by sending electronic invoices, new control indicators are required: e.g. log files and encryption proof instead of paper copies of the invoice.

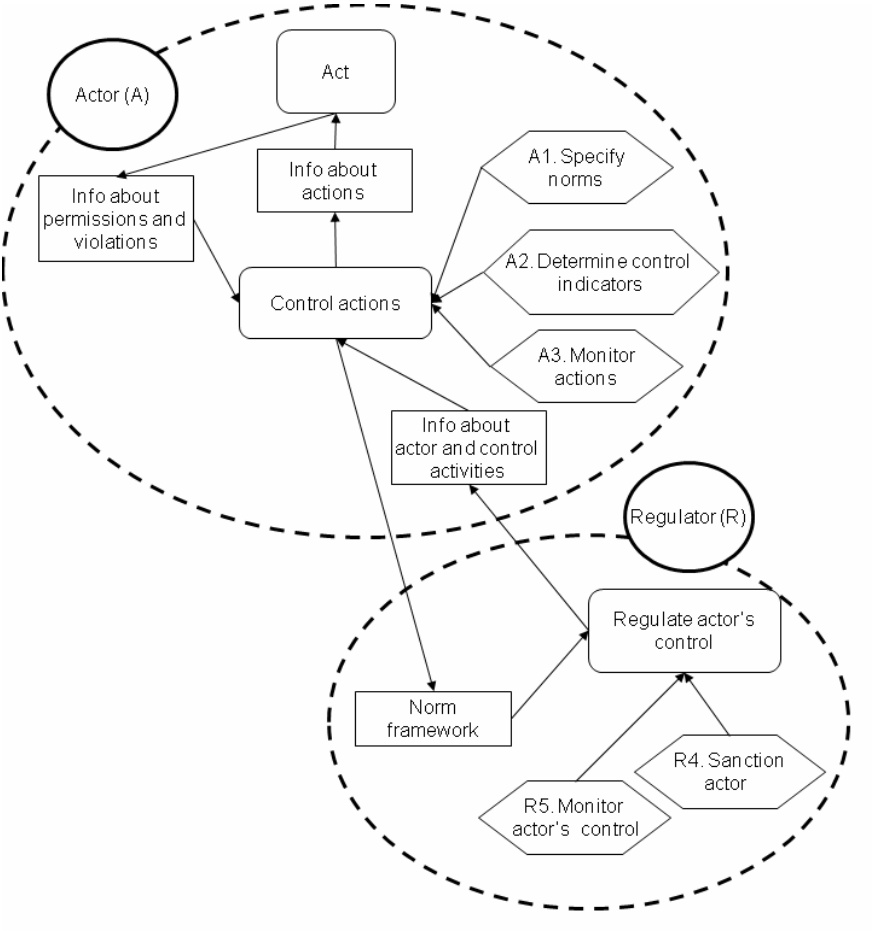


**Fig. 2.** TROPOS model of self-regulation, control tasks of the regulator are internalized by the actor agent

Now we describe the consequences of A's internalization of the control tasks of R's goals and plans. Since A now has to control its own actions, the goal of R to regulate A's actions is supposed to be met by the control activities of A. To determine if this delegation of control is effective, R has adopted a new goal which is to regulate the control activities of A. To reach this goal, R also has defined a new plan (R5), which describes the activities of R to monitor and evaluate A's control actions. Note that R now depends on A for information about its control activities instead of its activities, so R5 is a kind of meta-control plan. In auditing practice, R5 refers to a system-based audit, were the focus is the internal control system instead of on business transactions. Before an agent can enter a self-regulative relation, it has to provide an authenticated control architecture or control script to the regulator.

A similar solution would work for a code review, mentioned in the introduction. An electronic marketplace, for example, may want to provide assurance that its members generally comply with the norms. In addition to monitoring and sanctioning violations when they occur, the institution can require the owner of an agent to provide documentation and programming code before an agent is allowed into the environment. In this documentation the owner must make explicit how the agents internal control architecture assures compliance to the norms defined by the marketplace. Using this evidence, the institution can verify (up to a point) whether the agent is compliant by design, compare [17]. Such a verification can be automated, but that is not necessary, as the review takes place off-line.

Figure 2 shows dependencies between agents A and R engaged in self-regulation.

When we compare direct control with self-regulation we see that A internalizes some of R's control activities on A. New information resources are gathered to be used within the control activities. Also new goals evolve and lead to the adoption of new plans. Corresponding new dependencies between R and A develop for the acquisition of these new information resources.

These inter-agent changes are reflected in a number of objectives for intra-agent architectures. Summarizing the objectives, self-regulating agent must have at least the capabilities to:

1. Detect, internalize and store the norms which are applicable in the environment,
2. Translate norms into measurable control indicators, and
3. Monitor, detect and mitigate possible norm violations.

In the next section we show the internal architecture of the actor in self-regulation.

## 3 Inter-agent Analysis

We now analyze how the new tasks and dependencies revealed by the TROPOS models will affect an agent's internal architecture. We acknowledge that these tasks are complex normative tasks. We use the Intelligent Resource-Bounded Machine Architecture (IRMA) [4]. The architecture is a BDI architecture where the intentions are structured into plans. A plan can be a plan that an agent has actually adopted, or a plan-as-recipe that is stored into the plan library. Plan options are proposed as a result of means-end reasoning or by the opportunity analyzer. The opportunity analyzer

detects changes in the environment and determines new opportunities, based on the agent's desires. The options are filtered through a compatibility filter, that checks the options to determine compatibility with the agent's existing plans, and a filter override mechanism, in which the conditions are defined under which (portions) of plans need to be suspended and replaced by another option. The deliberation process determines the best option on the basis of current beliefs and desires.

Consider an autonomous agent that likes to achieve a certain goal. The agent has already several plans of action available (in its plan-library) to reach this goal. Before deliberating on a plan, the agent engages in a filtering process. This process constrains the agent's possible plans to plans that can be completed given its available (sub) plans in the plan library, its beliefs and desires. The agent chooses from this selection the best plan, given its beliefs and desires, and executes the plan. Figure 3 shows our extension of the IRMA architecture, adapted for self-regulation. Norm related adaptations are shown in grey and dotted lines. The ovals in the figure are information stores (repositories) and the rectangles are process modules.

Within IRMA we like to implement the processes and information stores that are needed for self-regulation. A self-regulating agent needs to internalize certain control activities to control its actions. The activities are: 'specify norms' (A1), 'determine control indicators' (A2), and 'monitor actions' (A3). These control activities require input from the agent's actions, and the actions in turn are influenced by the norms. We first analyze which IRMA modules are possibly affected by normative reasoning.

Norms can impact the information stores or processes of the architecture. A norm can be implemented in plans and function as a threshold to restrict the outcome. For example, a thermostat function that tries to keep the room heated at a certain temperature. Norms can also restrict the possible set of plans. Plans that violate a norm are no longer stored in the plan library. Or in means-end reasoning: there are illegal plans in the plan library but they are not considered as appropriate options to reach a goal; such plans are temporarily 'suppressed', as in [14]. Norms can also prevent the actual execution of a plan. For example, a person can plan to rob a bank, but decide not to do so at the last moment.

Besides that, norms affect the beliefs. After all, agents are expected to know the general norms. Beliefs also affect the norms, in the sense that beliefs about the context help to identify applicable norms. An agent may also realize, based on its beliefs, that it is acting in violation with the norms. Or, an agent realizes that due to a change in activities certain norms are no longer applicable and new norms must be incorporated. Whenever an agent adopts a new norm, this must be known (believed).

Norms are also related to the desires of an agent. An agent's desires may violate the norms. For example, an agent may desire a handbag that is made of the skin of a protected snake. A norm is that killing a protected animal is illegal. If norms are included in the compatibility filter, an agent can check if an option is compatible with its norms. If norms are part of the filter override mechanism, non-compliance can be a condition under which an agent must reconsider its plans. Both implementations make it possible for an agent to decide not to consider a plan option of buying a snake skin handbag. The opportunity analyzer may use the norms and beliefs to search for an alternative, such as a fake snake skin handbag.

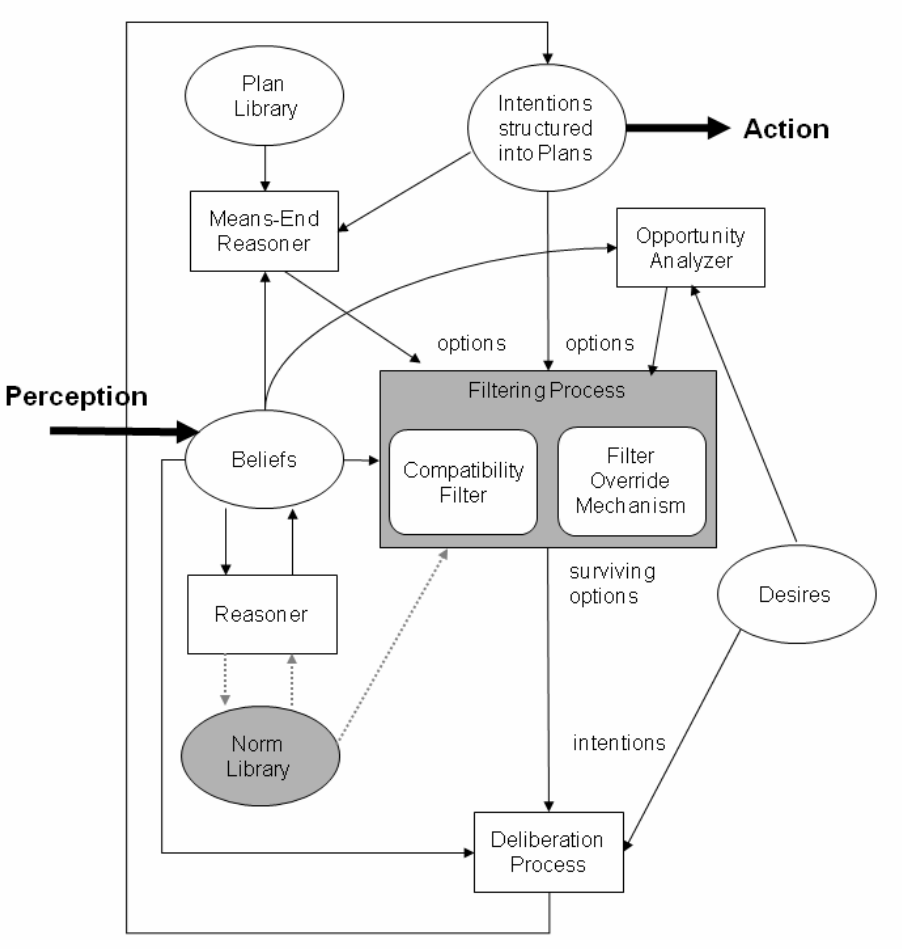


**Fig. 3.** A reasoning component for self-regulating agents adapted from [4]. Norm related adaptations are shown in grey and with dotted lines.

We find that norms can impact all components of the architecture. To assure consistent norm application we propose a central storage for norms similar to what the plan library is for plans. Activity A1 updates the norm library according to the beliefs of the agent. Only norms that are considered to be applicable to the agent's specific situation are included. To make an agent aware of a norm (violation) we connect the norm library with the reasoner module that is attached to the beliefs. If an agent then reasons about its beliefs, it will take the relevant norms into account. Beliefs about a norm (violation) can be used as input for the means-end reasoner, opportunity analyzer and the deliberation process. Besides that, the agent may use its knowledge about norms to determine the control indicators of A2. We consider the filtering process the best location to implement the control indicators. Beliefs about norms are already included in the other reasoning processes. The filtering process and reasoning

thus together consider (non-) compliant behavior. We think that the majority of the control indicators should be embedded in the compatibility filter and only severe violations should be handled by the filter override mechanism. Otherwise it could happen that the filtering is too strict. The monitoring in A3 is handled through a comparison of the beliefs about the data on the indicators with the norms. Based on results from this analysis, controls in the filtering process may be adapted. Figure 3 shows a version of the IRMA architecture specifically adapted for self-regulation.

Our approach of embedding norms into the filter override mechanism is comparable with the framework that is proposed by [14]. Norms can also be implemented into the goal generation mechanism as was done in the BOID architecture [6]. In BOID one can distinguish two kinds of goals: internal motivations (desires), representing individual wants or needs, and external motivations (obligations) to model social commitments and norms [6]. All these potential goals may conflict. To resolve conflicts among the sets of beliefs, obligations, intentions and desires, a priority order is needed. In the BOID, such a (partial) order is provided by the agent type. In [8] we discuss the use of values for goal conflict resolution.

Note that we have so far only considered adaptations to the agent architecture based on a conception of norms as a kind of filtering: actions, plans or goals which might lead to violation are filtered out or suppressed. Instead of filtering, we can also consider norm adoption [9,11]. Here, a norm is simply adopted as a goal. The rest of the architecture will then ensure compliance. Note that adopted norms will often correspond to so called maintenance goals: goals to make sure that some desirable state of affairs subsists or that an undesirable state is avoided, by contrast to achievement goals, which are about reaching a new state of affairs. Architectures for dealing with maintenance goals are discussed in [13].

## 4 Case Study: AEO Certification

We illustrate and validate our models by analyzing a specific case of self-regulation: AEO certification. The case study results are based on document analysis and a series of semi-structured interviews with experts from Dutch Tax and Customs Administration, held in the period of May till November 2009. Meeting notes were made by the authors and verified by interview partners. Intermediate results of the case study were validated in a one-day workshop with domain experts.

An Authorized Economic Operator (AEO) can be defined as a company that is in-control of its own business processes, and hence is considered trustworthy throughout the EU in the context of its customs related operations [12]. Typically, modern enterprise information systems (ERP, CRM etc.) play an essential role for companies to be in-control. AEO's will receive several benefits in customs handling, such as a "Green Lane" treatment with a reduced number of inspections. This can lead to considerable cost-reductions for businesses. For non-certified enterprises customs will continue to carry out the traditional supervision. Customs can direct their efforts towards non-certified companies to increase the security of international supply chains, while at the same time reducing the administrative burden for AEOs.

To qualify as AEO, a company must meet a number of criteria, which are described in the community customs code and the AEO guidelines [12], developed by the European Commission. Part of the application procedure is a self-assessment on

the quality of the company's internal control system for aspects that are relevant to the type of AEO certificate ('Customs simplifications', 'Security and safety' or 'Combined') [12]. The company's approach and the results of the self-assessment are inspected by customs. Customs officers determine whether the self-assessment is performed well and whether the results indicate that a company is able to control its business processes such that they contribute to a secure supply chain. If this is the case and other criteria are met (e.g. solvency, no known tax evasions etc.) an AEO certificate is issued by the customs office. Next we focus on the self-assessment task.

### 4.1 The Self-assessment Task

The company's first task is to collect information related to the specific nature of the company to focus the self-assessment. This step is called 'Understanding the business'. The next step is to identify (potential) risks to which the business is exposed using the AEO guidelines, which provide an overview of general risk and attention points. The company determines which sections are important according to the nature of the business activities. A company then has to identify, what risks affect the supply chain's safety, and are therefore of interest of the customs authorities. So the company takes over the customs' task of risk identification. For example, computer components are valuable goods, which are subject to theft. Trading valuable goods requires more security measures, than, say, trading in a mass product like fertilizer. However, some ingredients of fertilizer may be used to assemble explosives, leading to a different set of risks.

A company must then assess if appropriate internal control measures are taken to mitigate these risks. The vulnerability of a company to threats depends on its current control measures. Control measures either reduce the likelihood, by dealing with vulnerabilities (preventative controls), or reduce the impact (detective and corrective controls). A robust system of controls is thus able to prevent, detect and correct threats. A robust system of controls should also monitor its own functioning. For risks that are not controlled, additional measures may be implemented or the risk is "accepted". Risks can be accepted, if the likelihood of a threat is limited and the risk is partially covered, or if the costs for complete coverage are very high.

The company must show how its risk management contributes to its being trustworthy. In addition, the company must evaluate whether the proposed measures are implemented effectively. To provide some guidance on what is considered 'effective implementation' customs refer to the COSO internal control guidelines. COSO is a general framework for risk management and internal control [10]. The scores range from 0 "no control measures in place", 1 "internal control is ad hoc and unorganized", 2 "internal control has a structured approach", 3 "internal control is documented and known", 4 "internal control is subject to internal audits and evaluation" until 5 "internal control measures are integrated into the business processes and continuously evaluated". This scoring provides the customs with an indication of the maturity level of the company's self-controlling abilities.

### 4.2 Case Analysis

In the AEO case study we see the implementation of tasks A1, A2, and A3 at the company's side. A company has to define a control system appropriate to handle its

specific risks. The company therefore translates the general AEO guidelines into norms that are applicable in its own practice and circumstances (compatible with A1). Thereby a company determines parameters to monitor and control its business processes (A2). A company with a control system of a high maturity level monitors its actions (A3) through internal audits and controls that are integrated in the processes. The customs replace their traditional controls of the company's processes (R1,R2,R3) with an assessment of the company's self-regulating capabilities and control actions (R5). To check the reliability of the company's controls, customs may still take samples of business transactions, but these will now be much more focused, for instance on areas with an increased risk. To make R5 and the delegation of tasks A1, A2, A3 more manageable, we see that additional guidelines and principles must be formulated. An example of such additional guidelines is the set of AEO guidelines specified by the EU, indicating examples of risk areas for different domains. Another example is given by local customs directives. For instance, the Dutch customs refer to the COSO maturity levels as a way of objectively measuring 'effective implementation' of control measures. Such additional guidelines are needed for both implementing and auditing control systems. They help to specify under what circumstances a company can be said to be 'in control' of safety and security.

We also observe dependencies regarding information resources. The company depends on abstract norms (e.g. the AEO Guidelines) provided by the customs, which they try to apply to themselves as they believe the customs would do. The customs on the other hand depend on the company for information about their control system. For instance, why have they chosen for a certain implementation of the norms? Why have they decided to accept a certain risk, and not take additional control measures?

The AEO case provides us with a new approach to control that could also be applied to a multi-agent system. It shows that norm enforcement is a task that can be distributed among various types of agents. Furthermore we learned that self-regulation only works under certain conditions and that delegating control tasks is not simple. In general companies find it difficult to do a self-assessment as they do not know what customs expect from them (open norms). The translation of the abstract AEO guidelines into company specific norms turns out to be hard. For companies it is unclear when they have taken sufficient measures. Companies sometimes expect customs to indicate what is sufficient: "A fence for a chemical company should be X meters high". Even for customs such knowledge is often only implicitly available as expert knowledge that is difficult to externalize and make accessible for companies. In the AEO case, implemented measures are based on the risks in the environment. This corresponds to our observation in the architecture that norms depend on the beliefs. In the AEO case, we find both the adoption of new policies and procedures (norm adoption), as well as a redesign of existing business processes (norm filtering). Mature companies have their controls integrated in the business processes, and have regular audits to check the functioning of controls (reflective capabilities).

Summarizing, we can say that the internal control system of a company can be seen as the implementation of an architecture for self-regulation. In the AEO case, customs must provide a kind of quality assessment of this control architecture (system based audit), rather than verifying business transactions. This fundamental change in the role of customs, shows a transformation from operational control to meta-control. Therefore issues like trust and integrity now play a role at two levels. First there is the trustworthiness of the company's management, or in the case of an electronic institution, of

the agent owner. In the AEO case, we find that historical indicators of fraud always lead to a rejection of the certificate. In electronic institutions a rule could be that agents from proven untrustworthy owners are denied access to the community. Second, there is the reliability of the control system or agent architecture itself. If the control system is not reliable, it cannot be used to take over the delegated control tasks. Therefore the company can't function as a trusted partner of customs. Electronic institutions may also require that agent behavior is controlled to assure the correctness of the transactions. In that case, the owner of must prove that the agent is compliant by design. Such a proof depends on the internal architecture of the agent.

## 5 Discussion

We use a combination of TROPOS and IRMA as a means to identify requirements for self-regulating agents at the intra- and inter-agent level. We do not claim that these are the best approaches currently available, there are some limitations.

The most important limitation of IRMA as the internal architecture is that it is not reflective. By this we mean that agents cannot learn from their mistakes. When the agent finds that a plan leads to a norm violation it is only able to cancel this plan as a possible option. It lacks mechanisms to delete or change such plans in a plan library. Desires that violate norms cannot be changed either. The agent therefore keeps proposing violating plans and desires. Since norms are context dependent it is quite complex to differentiate violating plans from non-violating plans. Plans that are allowed in one situation may be a violation under different circumstances. An adaption of the planning mechanism is needed.

Second, there seems to be a fundamental problem in the delegation of control: often it is not clear how to communicate the delegated norms from the regulator to the actor. For companies it is difficult to interpret and implement the EU guidelines. Should customs and companies implement a communication protocol, a shared vocabulary or procedures such that they can more effectively communicate information? How should a company make its internal control system available to customs, such that they can determine the quality of a control system in a specific context with limited expert knowledge? These questions have to be answered by a detailed study of norm communication, both in practice, and for multi-agent systems.

Third, our case study reveals issues which may inspire the development of norm enforcement mechanisms for multi-agent systems. Open norms and forms of self–regulation enable heterogeneous agents to enter environments as the entrance requirements (open norms) leave room for specific implementations (control architectures). In particular, it suggests how agents can be verified to be 'compliant by design' before entering into an agent environment. Agents have to map their control architecture to the norm framework of the environment. The communication problem in the case emphasizes the need for well defined norm frameworks. Norm frameworks should be abstract enough to allow agents to enter an environment and specific enough to support compliance and enforcement of the norms. However, exactly how to translate such ideas depends on the particular set-up of the multi-agent system. In most cases, there will be a large role for the human owner of the agent, similar to the role of management. They must provide the evidence to demonstrate that they are 'in control' regarding compliance.

## 6 Conclusions and Further Research

In this paper we have argued that, with regard to norms, the macro-level definition of tasks and dependencies in a multi-agent system and the internal architecture of agents are crucially interconnected. A combined approach, that analyses the inter- (between agents) and intra-agent level (inside agents), was suitable to identify objectives for an architecture for self-regulation. We identified key processes and their influence on the dependencies between agents and the internal agent architecture. The models provide an insight in the differences in requirements for direct controlled agents and self-regulating agents. The analysis also points out the limitations of some well-known existing approaches. IRMA lacks reflective capabilities and is therefore not sufficient to model a truly self-regulating agent: an agent that is able to learn from its mistakes. In [8] we look in more detail at various cognitive architectures and how they account for compliance and norm adoption. Also unaddressed were aspects of norm communication. For two agents to engage in self-regulation relation, they must be able to communicate the norms effectively. Since the agents are autonomous we cannot simply assume that both agents use similar vocabularies or protocols [7]. So a solution for norm communication should take the agent's autonomy into account. One of our current research projects examines the use of argumentation theory [1] for norm implementation and communication. Future research thus concerns the role of reflection for normative behavior and norm communication. We hope to specify our ideas more formally in a declarative agent architecture (e.g. based on [13]). This will allow agent-based simulation of the regulatory process. We are also interested in the evolution process from direct control to self-regulation.

**Acknowledgments.** We would like to thank the Dutch Tax and Customs Administration for their cooperation and insights.

## References

1. Atkinson, K., Bench-Capon, T., McBurney, P.: Computational representation of practical argument. Synthese 152(2), 157–206 (2006)
2. Boella, G., van der Torre, L., Verhagen, H.: Introduction to normative multiagent systems. Computational and Mathematical Organization Theory 12, 71–79 (2006)
3. Braithwaite, J.: Enforced self-regulation: a new strategy for corporate crime control. Michigan law review 80, 1466–1506 (1982)
4. Bratman, M.E., Israel, D., Pollack, M.: Plans and resource-bounded practical reasoning. In: Cummins, R., Pollock, J.L. (eds.) Philosophy and AI: Essays at the Interface, pp. 1–22. MIT Press, Cambridge (1991)
5. Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., Mylopoulos, J.: Tropos: An agent-oriented software development methodology. Journal of Autonomous Agents and Multi-Agent Systems 8, 203–236 (2004)
6. Broersen, J., Dastani, M., Hulstijn, J., van der Torre, L.: Goal generation in the BOID architecture. Cognitive Science Quarterly 2(3-4), 431–450 (2002)
7. Burgemeestre, C.B., Liu, J., Hulstijn, J., Tan, Y.: Early Requirements Engineering for e-Customs Decision Support: Assessing Overlap in Mental Models. In: Proceedings of the CAiSE Forum, pp. 31–36 (2009)

8. Burgemeestre, C.B., Hulstijn, J., Tan, Y.: Agent Architectures for Compliance. In: Aldewereld, H. (ed.) ESAW 2009. LNCS, vol. 5881, pp. 68–83. Springer, Heidelberg (2009)
9. Conte, R., Castelfranchi, C.: Cognitive and Social Action. UCL Press (1995)
10. COSO enterprise risk management framework, http://www.coso.org
11. Dignum, F.: Autonomous agents with norms. Artificial Intelligence and Law 7, 69–79 (1999)
12. European Commission: AEO Guidelines, TAXUD/2006/1450 (2007), http://ec.europa.eu/taxation_customs/customs/policy_issues/
13. Hindriks, K., van Riemsdijk, M.B.: Satisfying maintenance goals. In: Baldoni, M., Son, T.C., van Riemsdijk, M.B., Winikoff, M. (eds.) DALT 2007. LNCS (LNAI), vol. 4897, pp. 86–103. Springer, Heidelberg (2008)
14. Meneguzzi, F., Luck, M.: Norm-based behaviour modification in BDI agents. In: Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'09), Budapest, Hungary, pp. 177–184 (2009)
15. Rao, A.S., Georgeff, M.P.: Modelling rational agents within a BDI-architecture. In: Principles of Knowledge Representation and Reasoning (KR'91), San Mateo CA (1991)
16. Rees, J.: Self-regulation: An Effective Alternative to Direct Regulation by OSHA? Policy Studies Journal 16(3), 602–614 (1988)
17. Sadiq, S.W., Governatori, G., Namiri, K.: Modeling control objectives for business process compliance. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) BPM 2007. LNCS, vol. 4714, pp. 149–164. Springer, Heidelberg (2007)
18. Yu, E.K.S.: Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering. In: Proceedings of the 3rd IEEE International Symposium on Requirements engineering, pp. 226–235 (1997)