

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

Brigitte Burgemeestre, Joris Hulstijn, and Yao-Hua Tan

**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.

**Index Terms**—self-regulation, agent architectures, norm compliance

## I. INTRODUCTION

To motivate autonomous agents to comply with norms various enforcement mechanisms have been proposed. Norms here define 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 [8]. Enforcement mechanisms often require the introduction of special “observers” or “regulator agents” that actively monitor the behavior of the other agents [1]. Such agents are assigned to monitor the behavior of 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). Such models aim to analyze agent interactions and dependencies to construct norm enforcement mechanisms. 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 from each other, especially in norm enforcement where perceptions of 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 elaborates 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/ or enforcement are carried out by the agent itself, instead of a regulator agent or institution. It can be an alternative or extension to direct control, when external supervision and norm enforcement are not possible at all, are ineffective or when there is a lack of controlling resources. For example, in e-institutions it might be impossible to check all agent actions for compliance in real time. A solution then might be to do a code review up forehand and determine if an agent is compliant by design. In human societies programs of self-regulation have been found to contribute to expanded control coverage and greater inspectorial depth [2]. Self regulation can be implemented in various ways: from voluntary self regulation, where a group of agents voluntary 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, to a combination of mandated self regulation and direct control by regulator agents [10]. Each model of self-regulation causes different agent dependencies and information needs, which imposes different requirements on the IT architecture.

A special case of self regulation for international trade is the Authorized Economic Operator (AEO) program [7]. The AEO program is a European wide customs initiative that aims to secure the supply chain while at the same time reducing the administrative burden for companies through the use of self-control. Companies that are reliable 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 effective self-control is an obligatory 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 a combined approach to develop an architecture to embed self-regulation as a control mechanism for multi-agent systems.

Manuscript received June 26, 2009. This research was partially funded by PGS IT Audit of the VU Amsterdam and the integrated project ITAIDE of the 6th Framework of the IST Programme of the European Commission

All authors are with the faculty of Economics and Business Administration of the VU University, Amsterdam, The Netherlands. The third author is also with the Dept. of Technology, Policy and Management of the Technical University Delft.

Brigitte Burgemeestre (e-mail: cburgemeestre@feweb.vu.nl).

Joris Hulstijn (e-mail: jhulstijn@feweb.vu.nl).

Yao-Hua Tan (e-mail: ytan@feweb.vu.nl).

In this paper we present our first steps towards an architecture for self-regulating agents. The research questions we like to answer in this paper are: 1. What objectives need to be met by an architecture on self-regulating agents? 2. How do we need to adapt existing Beliefs Desires Intentions (BDI) [9] architectures? As a starting point we propose a combination of frameworks to cover the inter- as well as the intra-agent analysis. For the inter-agent analysis the Intelligent Resource-Bounded Machine Architecture (IRMA) [3] is a good starting point because it is a general BDI architecture that is well accepted and has formed the basis for more recent agent architectures. Software engineering methodology TROPOS [6] provides suitable concepts to analyze and model agents' dependencies. We analyze direct regulation and self-regulation using TROPOS (Section II). Using this analysis we generalize the objectives for the internal architecture of a self-regulating agent. We try to embed the normative objectives in IRMA (Section III). Using the extended architecture and TROPOS model, we analyze a case study of AEO (Section IV). We examine if our adapted version of the architecture covers the findings of the case study. We identify its suitability and the shortcomings.

## II. INTER-AGENT ANALYSIS

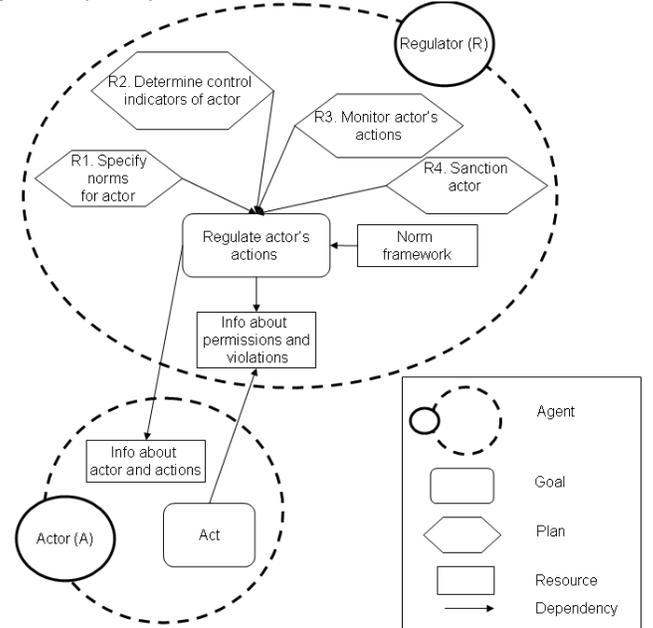
We first analyze the agents and the dependencies among agents. To do this we use concepts from the early requirements phase of the TROPOS methodology [4], which is derived from the i\*conceptual framework[11]. The key concepts we use are: actor, goal, plan, resource and dependency. An actor can be an autonomous agent that has a goal or strategic interest. 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 the direct control approach where the actions of autonomous agents are regulated by special regulator agents. After that we analyze self-regulation and assess what changes when an autonomous agent internalizes control tasks of the regulator agent.

### A. 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 has to have the following plans: 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 'control indicators' of A. 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 always make a copy of the invoice and store the copy to be able to check if the invoice payments are correct and complete. R3 is 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 only acts. 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 dependency analysis for direct control.



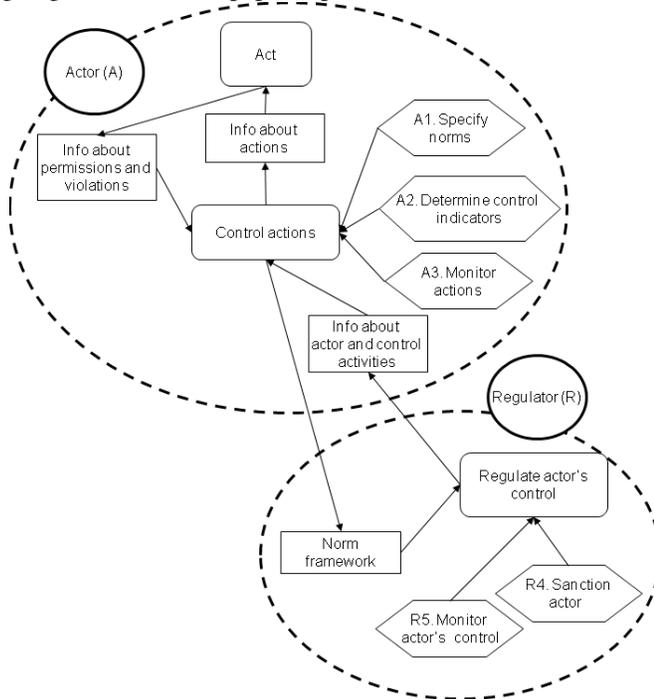
**Figure 1: TROPOS model of direct control. The actions of an actor (A) are regulated by a regulator (R). Note that arrows depict dependency, not information flow. So to regulate A's actions, R depends on A for info about actor and actions.**

### B. 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 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 motivate A to regulate itself appropriately. Furthermore to maintain the power of the regulator to handle non-compliant agents, the sanctioning task (R4) remains the regulators 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 compliantly with the norms. The acts of A in return 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 them electronic invoices, new control indicators are required; e.g. log files instead of paper copies of the invoice.



**Figure 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’s 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). R5 describes the activities of R to monitor and evaluate A’s control actions. R now depends on A for information about its control activities

instead of its activities. In auditing R5 refers to a system-based audit, where the focus is on the control system itself instead of the business transactions. Before an agent thus can enter in a self-regulative relation it has to provide for its authenticated control architecture or control script to the regulator. Figure 2 shows the dependencies between agents A and R when they engage 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 have to be gathered to be used within the control activities. Also new goals evolve and consequently the adoption of new plans. In correspondence new dependencies between R and A develop for the acquisition of other information resources

Summarizing, a self-regulating agent has to have the capabilities to: (1) Detect, internalize and store applicable norms in the environment, (2) Translate norms into measurable control indicators, and (3) ‘Monitor, detect and mitigate possible norm violations’. In the next section we zoom into the internal architecture of the actor agent in self-regulation

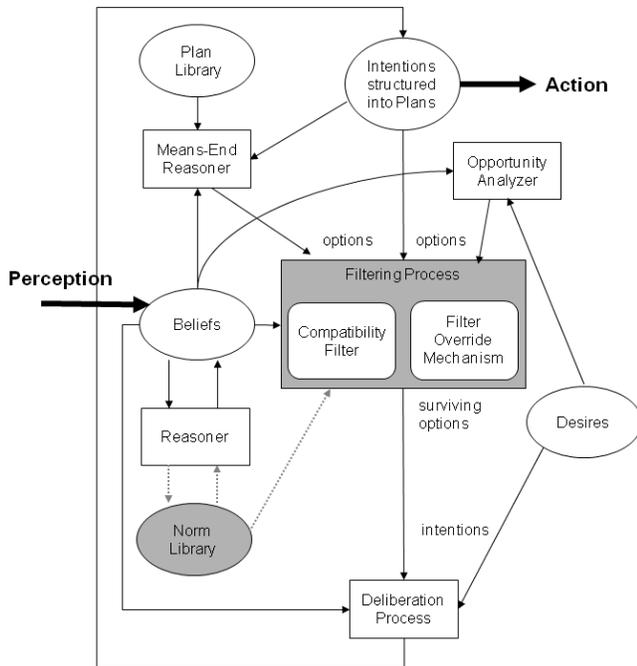
### III. INTER-AGENT ANALYSIS

We now analyze how the new tasks and dependencies revealed by the TROPOS models affect an agent’s internal architecture. We acknowledge that these tasks are complex normative tasks As a basis for our model we use the Intelligent Resource-Bounded Machine Architecture (IRMA) [3]. The architecture is a BDI architecture where the intentions are structured into plans. A plan can be the 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 what modules IRMA are possibly affected by normative reasoning



**Figure 3: A reasoning component for self-regulating agents adapted from [3]**

Norms can impact the information stores and 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 the norm are not stored in the plan library. Or in means-end reasoning: there are illegal plans available in the plan library but we do not consider them as appropriate options to reach a goal. 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.

Besides that, norms affect the beliefs, and beliefs affect the norms. An agent may realize, based on its beliefs, that it is acting non-compliant 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. When 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 always has to reconsider its plans. Both

implementations make it possible for an agent to decide not to consider a plan option that aims at 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.

We find that norms can impact all components of the architecture. To assure consistent norm application we propose a central information-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 takes the 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 an adapted version of the rational agent reasoning architecture for self-regulation.

Our approach of embedding norms into the filtering process is compatible with the framework that is proposed by [8]. Norms can also be implemented into the goal generation mechanism as was done in the BOID architecture [5]. 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[5]. All these potential goals may conflict with each other. To resolve conflicts among the sets of beliefs, obligations, intentions and desires, a priority order is needed. In the BOID, such a (partial) ordering is provided by the agent type.

#### IV. CASE STUDY AEO CERTIFICATION

We use our models to analyze 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.

An Authorized Economic Operator (AEO) can be defined as

a company that is in-control of its own business processes, and hence is reliable throughout the EU in the context of its customs related operations [7]. Typically, modern enterprise information systems (e.g 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. These benefits can lead to considerable cost-reductions for businesses. For non-certified enterprises customs will continue to carry out the traditional supervision. Customs can thus 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 [7], which are 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' [7]). The company's approach and the results of the self-assessment are inspected by customs. The customs 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 the other requirements are met an AEO certificate is issued by the customs office. Next we focus on the self-assessment task.

#### A. 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. The company thus replaces 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 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 has to motivate its choices in its system of control measures to customs. It has to show how its risk management approach contributes to being a self-controlling and reliable party. The company therefore evaluates the effective implementation of the proposed measures, using the COSO internal control scoring definitions. COSO is a framework for risk management and internal control [12]. 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.

#### B. Case analysis

In the AEO case study we see the implementations 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 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 replaces its traditional controls of the company's processes (R1, R2, R3) by an assessment of the company's self-regulating capabilities and monitors the control actions of the company (R5). We also observe dependencies on information needs. The company depends on abstract norms (e.g. the AEO Guidelines) provided by the customs, which they try to apply to themselves as customs would do. The customs on the other hand depends on the company for information about their control system.

The AEO case provides us a new approach of control that could be applied to a multi agent system. It shows that norm enforcement can be a task that can be distributed between 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 expects from them. Especially the specification of abstract norms of the AEO guidelines into company specific concrete norms proved to be hard. For companies it is thus unclear when they have taken sufficient measures to secure their part of the supply chain. Companies expect from the customs to indicate on a more detailed level 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.

When we look at the company's internal control system we see that norms have to be internalized based on perceptions of

the environment. Only applicable norms are implemented. The norms have to be implemented in a systematic and structured way such that they detect norm violations and prevent them from occurring. In the architecture we see norms implemented as a filtering mechanism. In the AEO certification we see norm control as a structured process. In addition, mature self-controlling companies may have controls integrated in the processes or audits to check the functioning of the controls. The total control system of a company could be seen as their implementation of the internal control architecture. Therefore these new monitoring activities of customs in the AEO case could be seen as quality assessment of such a control architecture rather than the traditional role of Customs to control the specific business operations of the company. This fundamental change in the controlling role of the government is often referred to as the transformation from operational control to meta-control, where operational control is delegated by the Customs to the companies themselves.

## V. DISCUSSION

The combination of TROPOS and IRMA for self-regulating agents also has its limitations. However, we do not claim that these are the best approaches currently available. Instead we used the approaches as a means to identify requirements for self-regulating agent at the intra- and inter-agent level. Below we describe the two most important limitations.

First, the most important limitation of the architecture is that it is not *reflective*. By this we mean that agents cannot learn from their mistakes. When the agent determines that a plan contains or leads to a norm violation it is only able to cancel this plan as a current possible option. It lacks mechanisms to delete or change such plans in a plan library. Desires that violate norms can also not be changed. 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 plan mechanism is needed.

Secondly, there seems to be fundamental problem in delegation of control; namely that often it is not clear how to communicate the delegated norms from the regulator agent to the regulated agent. For companies it is difficult to interpret and implement the customs' norms for their business activities. Should customs and companies implement protocols, a vocabulary or procedures such that they effectively can 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 and related questions have to be answered through a study of norm communication between agents.

## VI. CONCLUSION AND FURTHER RESEARCH

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 insight in 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 lacked in reflective capabilities and is therefore not sufficient to model a truly self-regulating agent: an agent that is able to learn from its experiences with norms and use these experiences as constraints for future normative reasoning. Also unaddressed were aspects of norm communication. For two agents to engage in a 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 [6]. A solution for norm communication should take the agent's autonomy into account.

Future research will zoom in on the role of reflection on normative behavior and the communication of norms. Besides that we are also interested in the evolution process of an agent from direct control to self-regulation.

*Acknowledgments* We would like to thank the Dutch Tax and Customs administration for their discussions.

## REFERENCES

- [1] G. Boella, L. van der Torre, and H. Verhagen: Introduction to normative multiagent systems. *Computational and Mathematical Organization Theory*, 12:71–79, 2006P
- [2] J. Braithwaite: Enforced self-regulation: a new strategy for corporate crime control. *Michigan law review* vol. 80, pp 1466-1506, 1982
- [3] M. E. Bratman, D. Israel, and M. Pollack. Plans and resource-bounded practical reasoning. In R. Cummins and J. L. Pollock, editors, *Philosophy and AI: Essays at the Interface*, pages 1--22. The MIT Press, Cambridge, Massachusetts, 1991.
- [4] P. Bresciani, A. Perini, P. Giorgini, F. Giunchiglia, and J. Mylopoulos. Tropos: An agent-oriented software development methodology. *Journal of Autonomous Agents and Multi-Agent Systems*, 8, pp. 203–236, 2004.
- [5] J. Broersen, M. Dastani, J. Hulstijn, Z. Huang and L. van der Torre. The BOID architecture - Conflicts between beliefs, Obligations, Intentions and Desires. In *Proceedings of the Fifth International Conference on Autonomous Agents*, pp. 9-16, Montreal Canada, May 2001
- [6] C.B. Burgemeestre, J. Liu, J. Hulstijn and Y. Tan: Early Requirements Engineering for e-Customs Decision Support: Assessing Overlap in Mental Models. In the *Forum Proceedings of the 21st CAiSE conference*, pp 31-36, Amsterdam, The Netherlands 2009
- [7] European Commission: AEO Guidelines, TAXUD/2006/1450, 2007. [http://ec.europa.eu/taxation\\_customs/customs/policy\\_issues/customs\\_security/aeo/](http://ec.europa.eu/taxation_customs/customs/policy_issues/customs_security/aeo/)
- [8] F. Meneguzzi and M. Luck: Norm-based behaviour modification in BDI agents. *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, Budapest, Hungary, 2009.
- [9] A.S. Rao and M.P. Georgeff: Modelling rational agents within a BDI-architecture, in *proceedings of Knowledge representation and reasoning (KRR-91) Conference*, San Mateo CA, 1991.
- [10] J. Rees, Self Regulation: An Effective Alternative to Direct Regulation by OSHA?, *Policy Studies Journal*, 16:3 pp 602-614, 1988
- [11] E.K.S. Yu, Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering, in: *Proceedings of the Third IEEE International Symposium on Requirements engineering*, pp. 226-235, 1997
- [12] COSO enterprise risk management framework. Available: <http://www.coso.org>