Agent Architectures for Compliance

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

[1] Faculty of Economics and Business Administration, Vrije Universiteit, Amsterdam
[2] Department of Technology, Policy and Management, Delft University of Technology
{jhulstijn,cburgemeestre,ytan}@feweb.vu.nl

Abstract. A Normative Multi-Agent System consists of autonomous agents who must comply with social norms. Different kinds of norms make different assumptions about the cognitive architecture of the agents. For example, a principle-based norm assumes that agents can reflect upon the consequences of their actions; a rule-based formulation only assumes that agents can avoid violations. In this paper we present several cognitive agent architectures for self-monitoring and compliance. We show how different assumptions about the cognitive architecture lead to different information needs when assessing compliance. The approach is validated with a case study of *horizontal monitoring*, an approach to corporate tax auditing recently introduced by the Dutch Customs and Tax Authority.

1 Introduction

A Normative Multi-agent System consists of autonomous agents who must comply to social norms [3, 13, 6]. Normative multi-agent systems are used to design electronic institutions [31], but can also be used to understand compliance schemes in human society. Interesting new forms of auditing and norm enforcement are *horizontal monitoring* [15] and *responsive regulation* [2, 4]. Such forms of auditing are based on mutual trust and understanding, and do not only monitor compliance behaviour, but also take the underlying motivation and corporate culture into account. This may lead to more robust compliance behaviour and can save all parties a lot of effort. However, it requires different kinds of evidence than a traditional command and control approach. What are the *information needs* for assessing compliance using such 'horizontal' approaches?

How a norm is adopted and followed by an agent depends on the cognitive architecture of the agent. By a cognitive architecture we mean a structured model of the various components or modules which generate the agent's behaviour. Different ways of formulating norms make different assumptions about the underlying cognitive capabilities.

For example, in the accounting profession there is a long standing debate between advocates of *rule-based* and *principle-based* ways of formulating norms. A principle-based formulation of norms assumes that agents can reflect upon the outcomes of their actions; a rule-based formulation of norms only assumes that agents can follow procedure. Therefore agents can 'hide behind the rules' and pretend not be responsible for the consequences: "... rule-based traditions of auditing became a convenient vehicle that perpetuated the unethical conduct of firms such as Enron and Arthur Andersen" [28]. The Sarbanes-Oxley act [27] was introduced in 2002 to avoid such accounting scandals. Although originally a principle-based norm, in practice it has been given a rule-based

interpretation. Our working hypothesis is that principle-based norms only work when a subject has adequate capabilities for norm adoption and compliance. Using cognitive agent architectures we hope to make such assumptions explicit. Sanctions form another example. The use of sanctions assumes that agents 'care' about the negative consequences and are actually able to adjust their behaviour. These assumptions are not true in all cases. For instance, artificial agents typically do not 'care'; only their owners do. Also, when agents lack expertise to improve their behaviour, sanctions are counterproductive. In such cases it is better to provide advice [2].

In this paper we investigate how agent architectures can account for compliance. We discuss several cognitive agent architectures taken from the literature. We show how the architectures can implement normative behaviour, namely by filtering or adjusting the data structures which generate behaviour (policy, plans or goals). Moreover, we show how different architectures lead to different information needs to assess whether an agent is compliant. Such information needs are required for the design of 'auditing tools' to support human auditors when assessing compliance of companies. Throughout the paper we will therefore compare agent architectures to practices in the business world. Another application is in the design of electronic institutions. There too, assumptions about the underlying agent architecture will influence compliance monitoring.

Consider an electronic service broker. Compare [31] for similar such applications. There are two kinds of artificial agents: service providers and service consumers. Agents are 'scripts' written in a programming language, which is run by the broker environment. The environment provides primitive actions for paying, communicating and delivering services. The purpose of the broker is to help consumers find providers and to help them negotiate, implement and maintain a service level agreement. Owners of participating agents pay a fixed membership fee and a percentage for each successful deal. The broker tries to make sure that (owners of) agents comply with the rules of the institution, i.e. follow the protocol and honour the service level agreements. This assurance is part of the added value of a broker. The broker must therefore monitor communications and transactions as they take place in the environment. Upon complaints or other evidence of fraud, a human investigation may be started. The main sanction is to evict perpetrators. However, sanctions always come after the fact. Therefore the broker should assess the compliance attitude of the owner of a participating agent before entrance, and do a code review of the scripts. Our information needs analysis can be used as a guideline for the set-up of such an assessment.

Our findings about information needs are illustrated and validated with a real life case study of *horizontal monitoring*, a form of auditing recently introduced by the Dutch Customs and Tax Authority (DTCA) [15]. The tax office relies as much as possible on the company's internal control system. Tax auditors only have to assess reliability of the company's internal control system, instead of the tax declarations themselves. This saves the tax office a lot of effort. To participate, the company must give tax auditors access to its records and policies. The tax office in return can give the company more certainty, for instance that a tax declaration for a preceding year is settled. The new audit approach requires different kinds of evidence and a different way of communicating.

The paper is structured as follows. In Section 2 we present three cognitive agent architectures for compliance. In Section 3 we outline the information needs for establishing compliance. In Section 4 we present the case study on horizontal monitoring.

2 Cognitive Agent Architectures

Below we discuss a number of general cognitive agent architectures. We determine what it means to be compliant given the capabilities offered by each architecture.

2.1 Perception and action

The most simple agent architecture consists of two modules: perception and action [26]. These are linked in a feedback-control loop (Figure 1). Therefore this architecture is called a *reactive agent architecture*. The agents behaviour is determined by a data structure called *policy* π: a set of situation-action pairs $s \rightarrow a$, which specify for each possible situation s, what appropriate actions a would be. The policy is non-deterministic: several actions may be possible in a situation. The perception module determines on the basis of observations what situation the agent is likely to be in. Perception may be wrong. The agent has a (limited) memory of the previous situations it has been in. These may be called its *beliefs*. The action module simply executes the action suggested by the policy. Not all actions will be successful.

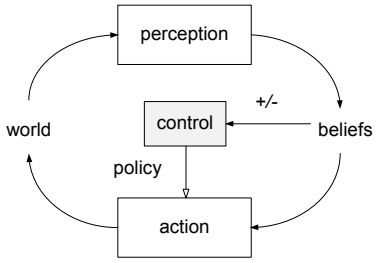

Fig. 1. Perception and Action

If we compare this to the business world, there are similarities with the way a company is run. The daily operations are usually determined by business processes and standardised operating procedures. Like the policy data structure, processes and procedures outline for known situations what to do.

What does it mean to be compliant given this architecture? In a reactive architecture, being compliant to a norm can in principle be implemented in two ways:

1. Adapt the perception module to send a warning signal when the next move is likely to lead to a violation, and to send an alert when detecting an actual violation. This is called *self monitoring*. Adapt the action module to suspend all actions after a warning and undo the previous move after an alert. This is called *self control*.
2. Adapt the policy data-structure and *filter out* all situation-action pairs which are known to lead to a violation. Depending on the computational complexity, such readjustment can be done either immediately after a warning or an alert, or off-line.

In an ideal world adjusting the policy is enough, but self monitoring and self control are necessary when, due to inaccurate beliefs or unsuccessful actions, the agent may find itself in or close to a violation despite the adapted policy.

The ability to monitor, evaluate and adjust future behaviour is crucial for being compliant. That means we need a *reflective agent*: an agent who has a representation of itself and can alter this representation in order to adjust its behaviour [22]. In Figure 1 this function is performed by a separate control module. Adjustments are indicated with an open arrow: $\multimap$. There are many ways to implement a reflective agent; see [29] for a survey. A very basic one would make use of *reinforcement learning*: adjusting the policy through signals from the environment [19]. After each action, the agent gets a reinforcement signal r. Violations receive a punishment ($r = -1$); other states get a reward ($r = 1$). The agent must optimise its policy π to increase the sum of reinforcement signals over a certain period. Many algorithms can solve such optimisation problems [19]. We mention a simple one as illustration (linear reward-inaction algorithm). Let p_i be the relative likelihood of taking action a_i in some given situation s according to the policy, and let α be a learning rate ($0 \leq \alpha \leq 1$). For all actions a_i, a_j which are possible in s, if action a_i succeeds ($r = 1$), then $p_i := p_i + \alpha(1 - p_i)$ and $p_j := p_j - \alpha p_j$ for $j \neq i$. When action a_i fails ($r = -1$), p_j remains unchanged for all j.

2.2 Perception, planning and action

The second architecture contains an additional module: planning. This module determines by what sequence of actions the agent can best achieve its goals. Therefore, this architecture is also called a *deliberative architecture*. A well known example is the BDI architecture [25]. The planning module takes two kinds of input: (1) the information produced by the perception module about the current state the agent is likely to be in, enriched with background knowledge on how to interpret situations. Again, this may be called the *beliefs* of the agent. (2) The *desires* of the agent, i.e. preferred future states of affairs, which function as potential goals. Desires may be mutually incompatible. The output of the planning module is the set of goals the agent has decided to pursue, and for which a particular plan has been adopted. These adopted goals are called *intentions*. Intentions must be mutually compatible. Following Pollack [24] and many subsequent BDI architectures, we suppose each agent is equipped with a plan library with useful *recipes*: pre-stored plans for different kinds of situations. When there are no recipes for a particular goal, a new plan must be generated from first principles.

In the business world, goals would correspond to objectives of an enterprise. Desires correspond to long term objectives, such as increasing market share or reducing time-to-market. Intentions can be compared to those objectives which have been aligned in a consistent strategy and for which processes, procedures and projects have been implemented. Plans or recipes correspond to business processes and procedures (for routine plans) or to projects (for one-off plans).

Again, such architectures are reflective: agents must be able to learn and adjust their behaviour. So when a plan fails to achieve its goals, it must be dropped or adjusted along with the plan library. This function is again performed by a separate control module. Desires or goals define the success of a plan, similar to the reinforcement signals in the previous section. The control module therefore takes goals as input.

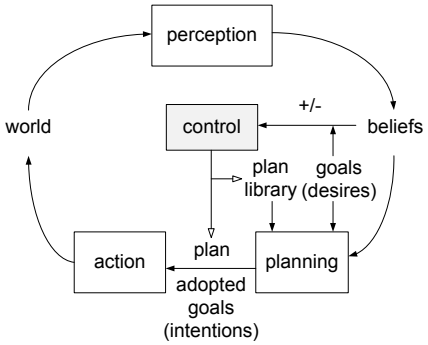

Fig. 2. Perception, planning and action

When we compare such reflective capabilities to the business world, the so called Deming Cycle comes to mind: *plan-do-check-act* [12]. Variants of this management control cycle are very influential in management approaches to quality control, risk management or information security. *Plan* and *do* correspond to our planning and action. *Check* (or study) refers to all kinds of learning and evaluation. This corresponds to our perception. *Act* refers to making decisions: altering the plan, based on the outcome of the evaluation. This is modelled here by the control module. Learning is also one of the cornerstones of capability maturity models [17]. In other words: mature organisations have the ability to learn, evaluate and adjust their behaviour.

What does it mean to be compliant in a deliberative architecture? Again, we can adjust execution by adding self monitoring and self control to the perception and action modules, or we can adjust the data-structures which generate the behaviour. The policy data structure is now replaced by goals and plans. In general, there are two ways of adjusting plans and goals to address compliance: by *filtering out* potential violations from existing plans and goals [23], or by *adopting a norm* as a new goal [6, 8, 13].

Filtering We can deal with compliance in three different ways: (1) filter the plan library (no capability to violate), (2) filter the adopted plans and goals (no intention to violate), and (3) filter the potential goals (no desire to violate). Filtering the plan library (option 1) can be done off-line. It needs an algorithm to scan all recipes for potential violations. Such recipes are then *suppressed*: they receive a special status such that they will never be adopted. This is essentially the compliancy approach of Meneguzzi and Luck [23], implemented in the AgentSpeak programming language.

Option 1 assumes that possible violations can be detected out of context. But in general violations are context dependent. Consider an agent intending to “go as fast as possible given the condition of the terrain”. We can’t decide off-line if this plan would violate a speed limit of 30 km/h. Or consider a budget of $20. Buying fuel ($8), tires ($11) and a new mirror ($3) will add up to a violation, but it is unclear which goal constitutes ‘the’ violation. Therefore potential violations need to be assessed in conjunction with current beliefs and relevant other goals. This can be done under option 2. Option 3 suffers from a similar problem: potentially illegal desires cannot be detected off-line.

Norm Adoption Instead of filtering, we can also produce compliant behaviour by adopting a norm as a source of potential goals similar to a desire. This approach is developed by [9], see also [6, 13]. Adopted norms will often correspond to *maintenance goals*: goals to make sure that a 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. For example, our agent could adopt a maintenance goal to obey the speed limit. A cognitive architecture which can deal with maintenance goals contains a kind of feedback-control loop [16]. Similar to what we discussed in Section 2.1, trigger conditions will produce a warning or an alert when the goal is likely not to be maintained (self monitoring), which will lead to the blocking of current actions or repair actions being executed (self control). Moreover, any new goal about to be adopted must be filtered, to verify whether it does not conflict with current maintenance goals.

Norm adoption will often lead to conflicts. Consider an agent who is late for an appointment and must drive to as fast as possible. Which goal should get priority: keeping the appointment (achievement goal) or obeying the speed limit (maintenance goal)? Often such conflicts will have to be resolved by the designer of the system; priorities are build-into the agent. Our next architecture can deal with conflicts explicitly.

2.3 Perception, goal generation, planning and action

The third architecture adds a separate module for *goal generation*. Goal generation was introduced by Thomason [32], who observed that until then in AI, goals were simply given. Thomason realised that even before planning, it makes sense to filter goals. Goals which are incompatible with current beliefs are impossible to achieve and may lead to *wishful thinking*. Such goals should never even be generated. The idea was taken over by the BOID architecture, which distinguishes between 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. New potential goals may also conflict with previously adopted goals (intentions). To resolve conflicts among the sets of beliefs, obligations, intentions and desires, some sort of priority order is needed. In the BOID, such a (partial) ordering is provided by the *agent type*. For instance, a so called *realistic* agent values beliefs over any kind of goals: $B > D, B > I, B > O$. This avoids wishful thinking, overcommitment or dogmatism. A *stable agent* values previously adopted goals (intentions) over new goals (desires; obligations): $I > D, I > O$. A *selfish* agent values its desires over social obligations, $D > O$, whereas an *obedient* agent will value obligations over desires: $O > D$. These orderings produce caricatures of agent behaviour. In practice, more detailed distinctions will be necessary. For instance, a stable agent might make an exception for life-saving obligations. The priority should be based on a classification of goals based on their relative value for the agent. In the business world, such a classification is often done by a risk assessment [10]

Again, the architecture is reflective: it can adjust its behaviour based on an evaluation of its relative success. What does it mean to be successful? For actions and plans, success is defined in terms of the underlying goals (desires, obligations). For selecting one goal over another, success depends on the agent type. Agent types are basic; they have no external motivation.

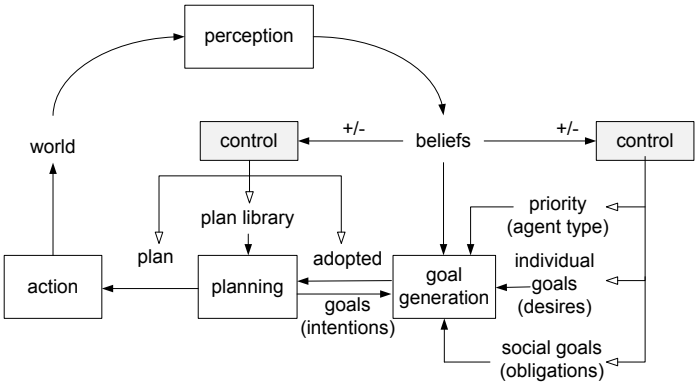

Fig. 3. Perception, goal generation, planning and action

What does it mean to be compliant in such an architecture? All previously mentioned ways of being compliant will work here too: self monitoring and self control, as well as adjusting the plans and goals or even the priority order guiding decision making. Given that resources (money, energy, time) are limited, adopting a norm as a goal means that other goals can no longer be achieved. Norms may conflict with individual desires, with reality, with previous intentions or with other norms. So norm adoption creates dilemmas; it requires a form of conflict resolution. Because agents are autonomous, we cannot ensure an agent will always adopt a norm. At least the agent should be aware of the consequences of possible violations such as sanctions or loss of reputation. When an agent nevertheless decides to violate a norm this should be a conscious decision.

How can dilemmas be resolved? A priority order determines how an agent will make choices. In the BOID, priorities are fixed by agent type. In general, the relative importance of goals is linked to *social values* [30]. In the business world, consider values like profit, safety, or quality. Such values are embedded in the corporate culture. For example, a culture which values short term profits over security – as apparent from payment and incentive schemes – will be more likely to lead to behaviour which violates a security norm, than a culture which values security over profits.

Social values may look vague, but fortunately, there has been progress in the computational representation of argumentation systems based on social values, see Atkinson et al [1]. Decision making is a form of *practical reasoning*. Crucial is what counts as a justification for an action. In a BDI architecture justification is essentially based on goals.But how are goals justified? Social values account for the fact that people may disagree upon an issue even though it would seem to be rational [30]. That means that justification becomes a matter of debate. Like in legal theory, it doesn't matter who is right; what matters is who can construct the most convincing argument and defend against counter arguments. An argument starts with a position [1]: "In the current circumstances s, we should perform action a, to achieve consequences s', which will realise some goal g, which will promote some value v." A position can be attacked by critical questions, which seek to undermine the underlying assumptions (Table 1). Critical questions provide a kind of quality test for the justification of a decision.

CQ1 Are there alternative ways of realising consequences s', goal g or value v ?
CQ2 Is it possible to do action a?
CQ3 Would doing action a promote some other value ?
CQ4 Does doing action a have a side effect which demotes the value v, or some other value?
CQ5 Are the circumstances s such that doing action a will bring about goal g?
CQ6 Does goal g promote value v? Is value v a legitimate value?
CQ7 Is goal g achievable?

Table 1. Critical Questions (Atkinson et al [1])

3 Information Needs

In this section we are concerned with the assessment of whether an agent is compliant or not. As we have seen compliance can be realised in many different ways. To demonstrate compliance to a norm, both subjects and auditors therefore need different kinds of evidence, based on the underlying cognitive architecture.

1. *perception, action* First, evidence of the behaviour itself, i.e. original records, data and log-files testifying that violations did not occur. This is the only way to show effectiveness of monitoring. Second, evidence of the policy avoiding violations, and evidence showing that the policy is instrumental in actually producing behaviour. This can be a review of the policy specification, or data-mining techniques discovering patterns of behaviour to indicate that there is a consistency in behaviour which suggests a policy is followed. Third, evidence of changes in behaviour after sanctions or rewards were administered. Again, this would be pattern detection, showing that certain non-compliant patterns ease after the sanctions were applied.
2. *perception, planning, action* In addition to the kinds of evidence listed above, for deliberative agents we can also look for evidence of plans and goals being compliant. Having such higher cognitive attitudes makes agents' behaviour predictable even in unforeseen situations. As evidence we need documentation on goals, plans and evidence that these goals plans are actually implemented and effective in producing behaviour.
3. *perception, planning, goal generation* In addition to the kinds of evidence listed above, now we also look for evidence of the agents' priorities in decision making: are obligations preferred over desires? Moreover, it must be assessed whether these preferences are used for day-to-day decisions and do not comprise a paper reality. Addition evidence may concern embedded social values.

How can such evidence be collected in practice? For all architectures, evidence of behaviour is needed. This is the only way to demonstrate effectiveness of the agent's internal monitoring and control. In auditing, such evidence corresponds to statistical samples from a pool of relevant events recorded in the company's information systems or log-files. Note that taking samples requires that the company is collaborating. In an electronic institution such as the service broker discussed in the introduction, we expect that log-files and traces of interaction and (relevant aspects) of transactions are accessible for monitoring by the institution.

For the more complex architectures, evidence about the cognitive 'data structures' which generate behaviour can be used in addition to evidence of behaviour. In practice, that means evidence of policies and procedures. In auditing theory, one distinguishes three audit aspects: design, implementation and operational effectiveness [18]. The *design* of a policy or procedure should guarantee that it will prevent, detect or correct undesirable behaviour. Evidence of *implementation* of a policy or procedure checks whether it has been adopted by the board, and whether employees know about it. Finally, evidence of *operating effectiveness* of a procedure should establish whether the procedure was followed in all relevant cases for the period under investigation. Note that this differs from the usual notion of effectiveness, i.e. meeting objectives. In an electronic institution, these aspects also make sense. Auditing the design would correspond to analysing the agents' specification. Auditing implementation corresponds to verifying the way agents have been programmed. Also the security of the electronic environment matters, whether it prevents unauthorised modifications to agent scripts. Auditing operating effectiveness corresponds to analysing log-files of agents behaviour. It also depends on the continued security of the electronic environment.

To summarise, we can demonstrate compliance by:

- *evidence of compliant behaviour*. This can be established by logging and monitoring the behaviour itself. In auditing, typically representative statistical samples are taken from a set of records of events. Logging and monitoring are greatly enhanced by tools for data mining and pattern recognition.
- *evidence of norm implementation*. This can be established by reviewing the policy or the plans in a plan-base. In a company this amounts to interviews, dossier research and reviews of documentation or board meeting minutes with the purpose of verifying the existence of relevant business processes and standard operating procedures. Implementation of a procedure is typically verified by a so called line control: for one or two representative cases, follow the procedure from the "cradle to the grave" and verify whether all relevant steps have been taken. In addition, one needs to establish whether the procedure has been operationally effective for the whole period (i.e. was not temporally switched off). This can be done by selecting a representative sample of cases (e.g. incidents or events), to verify whether in all those cases the procedure has been followed.
- *evidence of norm adoption*: This is hardest to establish. It means that norms have affected decision making. Presence of adopted norms in the form of objectives can be established by a dossier review combined with interviews with management (tone at the top). For those procedures, processes and projects implemented to meet these objectives, we must then verify the appropriateness of their design: would they indeed meet the objectives generated by the norm? Critical questions, similar to the ones mentioned by Atkinson et al [1] may be used to challenge the assumptions of a decision. Also the corporate culture may be a factor to support true norm adoption.

Discussion One may wonder why all of this is needed. After all, evidence of an agent's behaviour should be enough to determine whether the agent is compliant with the system norms. There are some reasons that evidence of norm adoption and norm implementation may be preferred. First, determining compliance through monitoring behaviour may take a lot of effort. In particular, for tax monitoring, this would involve large

statistical samples from past behaviour (See Section 4). In the 'horizontal' approaches some of this effort is delegated to the subject of the norm. Second, log-files can establish that behaviour is compliant, but not why. By contrast, compliance enforced through self regulation is said to be more robust [2]. So it makes sense to collect evidence about the way in which norms have been adopted and implemented. Third, monitoring behaviour can only lead to sanctions after the fact; there is no way of providing assurance beforehand. For example, in the case of the electronic service broker, mentioned in the introduction, the broker needs to provide some kind of assurance that agents will not violate norms. This can be done by an assessment of the intentions of the owner (evidence of norm adoption), and a code review (evidence of norm implementation).

Principle-based versus rule-based What can we say about the relative merits of principle-based or rule-based formulations of a norm? To make a comparison, we first need a concrete example of a principle and a rule. Consider the auditing principle of 'need to know': "only those people who need access to a document for their work, should be given access". When applying the principle to a specific situation, detailed models are needed of the organisational roles assigned to people, the tasks assigned to organisational roles, and the information needs associated with those tasks. In large enterprises such models are often unavailable or out of date. When they are available the models may be too restrictive, because there is a hidden trade-off between security and flexibility of the organisation. By contrast, a rule-based version of access control measures, would give a predefined checklist of those access rights which are not allowed for many common organisation roles. For instance, database administrators are not allowed to access the content of the database they manage. The problem is that a large part of such checklists is not relevant: there may not be any database administrators. Moreover, rules can be either too lenient or too strict for the situation. Still, rule-based norms are much more easy to implement and monitor.

It turns out that rule-based norms can be followed by all kinds of agents, including reactive agents. Principle-based norms, however, can only be followed by deliberative agents. When a principle is interpreted as a general (declarative) goal, which must still be adapted to the context, some form of planning is needed. When the principle involves a trade-off or dilemma, some form of ethical decision making is needed.

Sanctions What can we say about the effectiveness of sanctions as a way of enforcing norms? For reactive agents, it is important that sanctions follow immediately after the action that caused a violation. If there is a delay, the agent will not be able to adjust the relative weights for the right action in the policy data-structure. For deliberative agents, the goals (desires) are used to evaluate the success of an action or plan. Based on this evaluation, plans may be adjusted. So to have an impact, sanctions must actually be undesired; rewards must actually be desired. Sanctions are not necessarily monetary fines. For example, for many companies, certainty about the financial situation is more important than the possibility of a fine (Section 4). Empirical research among humans has shown that external sanctions can be counterproductive [14]. In a number of day care centres, they studied the introduction of a fine for parents being late to pick up their kids. After the fine was introduced, more parents were late. Apparently, they interpreted the fine as a kind of price, which made it morally acceptable to be late: the fine

removed their guilt. Removing the fine never restored original levels. For such reasons, non-instrumental motivations for following a norm, such as recognition of the authority, may lead to more robust compliance behaviour [7]. This is also one of the motivations behind responsive regulation approaches [15, 2]. Finally, sanctions assume agents have the ability and expertise to improve their behaviour. But not all architectures have capabilities for generating a plan, say, from first principles. Similarly companies often lack expertise concerning compliance issues. So even if they have the ability to change the do not know in which direction to change. Also in such circumstances, sanctions may be counterproductive. It is much more effective to provide guidance and advice on how to improve compliance [2].

4 Case Study: Horizontal Monitoring

Horizontal monitoring is introduced as the new compliance approach for tax audits concerning company taxation by the Dutch Customs and Tax Authority (DTCA) [15]. Companies which have shown that they have a reliable 'tax control framework', may voluntarily enter into horizontal monitoring. This is an auditing approach based on a relationship of mutual trust and understanding. Consider the following text from the brochure issued by the tax office [33]:

> "Horizontal monitoring entails mutual trust between tax payer and the Tax and Customs Administration, indicating more clearly everyone's responsibilities and abilities in order to do what is right, as well as laying down and observing reciprocal agreements. Horizontal monitoring is in line with developments in society, where the individual responsibilities of corporate and government managers and administrators are defined more clearly and upheld through supervision. Businesses must be transparent for stakeholders about the degree to which they achieve operational targets and the extent to which they are in control of the processes involved. The government is an example of a stakeholder."

Data Collection Our case study is in an initial stage. We are exploring hypotheses concerning changing information needs. The following findings concerning the tax control framework and the horizontal monitoring philosophy are based on two interviews with experts of horizontal monitoring within Dutch Tax and Customs Administration, as well as on publicly available policy documents [33]. Moreover, as part of previous research we have conducted several interviews with experts on AEO self-assessment, a compliance scheme used in customs, which is also based on Horizontal Monitoring.

The general idea of horizontal monitoring is that the tax office relies as much as possible on the company's internal controls. This is illustrated by the audit layer model of Dutch Tax shown in Figure 4. The model is nicknamed the 'onion model' because it represents auditing effort as consecutive layers which can be peeled off. The kernel is formed by the business processes. Reliability of the business processes is mostly established by the internal control framework. Reliability of the internal control framework is maintained by the internal audit department of the company. The internal controls are audited yearly by the external accountant as part of the financial audit. This means

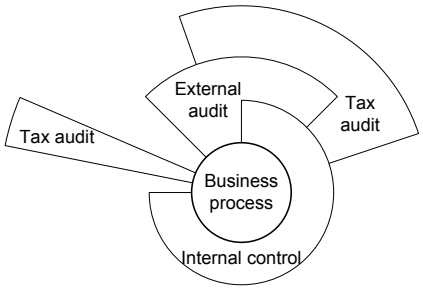

Fig. 4. Audit layer model (Dutch Tax and Customs Administration)

that the tax office now only has to assess reliability of the company's internal controls, internal auditing and external auditing, rather than having to establish reliability of the original records and business processes underlying the tax declarations. Only for specific tax issues do tax officers need to make visits and collect full evidence. To facilitate such 'meta-auditing', the company must provide access to all kinds of information about business processes and internal controls. The tax office in return can give the company more certainty, for instance that a tax declaration for a certain year is settled.

Mutual trust and understanding may sound nice, but there are real changes in the way companies are being treated. Below we list a number of illustrative changes.

1. *Demonstrating to be 'in control'* When entering into a horizontal monitoring relationship, the tax office must establish the reliability of the company's internal control framework. This forms the basis for their 'trust' in the company's record keeping. COSO provides a well known standard for setting up an internal control framework [10]. The standard recommends: (1) a control environment where integrity and ethical values are supported from the top management throughout the organisation. (2) risk assessment is performed to identify and manage risks relevant to the organisation. (3) Control activities such as policies, procedures and processes are implemented to ensure a company carries out management directives. Examples include approvals, verifications, reconciliations, reviews of operating performance, security of assets and segregation of duties. (4) Relevant company data contained in the information system should be communicated in the organisation and to the relevant stakeholders. (5) Ongoing monitoring to assess the quality of a companys internal control systems. When a company has implemented a control framework like COSO it can demonstrate to the tax administration that it is in control of its business processes.
2. *No hunt for mistakes* Under the tax control framework the tax office will not try and find as many mistakes as possible in a company's tax declaration; given the complexity of the legislation this is not difficult to do. Confronting a well meaning company year after year with the mistakes they have made, is counterproductive. Instead the tax office will try to help companies avoiding such mistakes in the future. This involves clear communication about what is expected of a company.
3. *Open communication.* Another issue regards the interpretation of the tax code. The Dutch corporate tax – by design – contains some space for interpretation. Compa-

nies are allowed to interpret the rules in their favour. However, grossly unreasonable interpretations are not considered acceptable. What is considered acceptable, differs from case to case. Currently, this creates uncertainty for companies. Under the new tax control framework, companies may seek advice with their tax office, to find out in advance whether their interpretation is considered acceptable. Such advance information sharing can save both parties a lot of work.

4. *Up to date tax assurance*. In corporate taxes it is customary to audit retrospectively. The tax officials are entitled to look back in the records to check for tax violations. A company can be fined by the tax administration for an error that occurred a few years ago. Resolving disputes is a long process resulting in high costs for both parties. In horizontal monitoring the company and the tax office agree to solve all historical issues when they start the relation. New issues are supposed to be solved immediately when they occur. A company no longer needs to worry about past tax issues and is assured that sanctions for historical violations will not be imposed.

These examples show that a relation based on mutual trust and understanding imposes requirements on the interaction between both parties as well as on the internal processes of the company. A company can demonstrate compliance by the implementation of an internal control system and open communication of the results. We show how different assumptions about the cognitive architecture(internal control system, ability to differentiate between compliant and violating behaviour, embedded plan-do-check-act cycle) lead to different information needs (emphasis on advice on norm interpretation instead of norm violations, new and timely data instead of historical data, information on implementation of norms, control procedures) when assessing compliance (principle based instead of rule based). We thus observe that a more mature agent architecture is needed to handle the new control approach of Dutch tax.

As part of the case study, we have looked at the internal auditing guidelines used by Dutch Tax to instruct auditors on the tax control framework [33]. It turns out that auditors are instructed to use evidence regarding decision making and implementation of decisions, in addition to the usual transaction based evidence. To demonstrate compliance, evidence about the design, implementation and operational effectiveness of cognitive data structures like procedures is needed. In practice this means that a company should provide tax auditors with detailed process descriptions, working procedures as they are implemented, and evidence that the procedures are known and applied.

5 Related Research

There is a growing body of research about compliance in various fields. We can only mention a few works that may be of interest.

In the field of multi-agent systems, the work by Meneguzzi and Luck [23] is similar to our work. They understand compliance as a filter of the plan library, implemented in AgentSpeak. The norm adoption approach has been advocated both Conte and by Dignum [8, 9, 13]. These architectures have influenced the BOID architecture of Section 2.3. Moreover, we believe that any architecture for norm adoption should be able to handle maintenance goals, see Hindriks and Van Riemsdijk [16].

Regarding reflective programming, there has been progress on systems for automated reconfiguration [11]. This work focuses on mechanisms for monitoring and diagnosis, in order to reconfigure its activities. They also need a declarative specification of objectives (i.e. goals) to evaluate the 'success' of an activity against.

Compliance can often be guaranteed by the design of business processes. Lu et al [21] discuss how to use business process management systems (BPMs) to automatically monitor and enforce compliance to norms and standards. Like us, this work combines formal reasoning and the agent metaphor with the actual practice of companies.

Finally, there is an interesting parallel between our information needs and the work of Lewicki and Bunker [20] who distinguish three sequential stages of trust. *Calculus-based trust* is based on the consistency of behaviour and involves a continuous evaluation by the actors of the punishment for violating trust and the rewards for preserving it. *Knowledge-based trust* occurs when one has enough knowledge about the other party's needs and preferences to understand them and to predict their likely behaviour. *Identification-based trust* is based on identification with the others' desires and intentions. There is a mutual understanding and appreciation of each others' wants, such that parties are able to act to the benefit and on behalf of the other.

One would expect a correspondence between the information needs for trust establishment and for assessing compliance. There are indeed many similarities, but a full mapping is not possible. For instance, Lewicki and Bunker put adoption of the other's intentions and desires under identification-based trust, where we would expect this to be part of the goal-based model. Social values or cultural cues, which we would expect under identification-based trust, are not mentioned by Lewicki and Bunker.

6 Conclusions

Compliance of agents has mostly been discussed at the inter-agent level; however, the intra-agent level is also important. After all, different kinds of norms make different assumptions about the cognitive architectures of the agents who must comply with the norm. In this paper we have tried to make such assumptions explicit by presenting three cognitive agent architectures from the literature and by indicating in which ways compliance can be implemented in these architectures. The architectures are: perception-action, perception-planning-action and perception-goal-generation-planning-action. All of these architectures also require reflective capabilities: to learn, evaluate and adjust behaviour based on experience.

Conceptually, compliance can be implemented either

- by a filter, restricting cognitive 'data-structures' like policies, plans and goals so that no violations will occur, or
- by norm adoption, where the norm itself is added as a goal or plan, so that the rest of the agent architecture will ensure execution.

The cognitive architecture of an agent has an impact on the evidence needed to assess whether the agent is compliant.

- For reactive architectures, all that is needed are log-files of the behaviour, to demonstrate that the agent behaves consistently, manages to avoid violations and seek preferred states.

– For goal-based architectures, also evidence of the plans and goals may be used in establishing compliance. In particular, when an agent states that it has as a goal to be compliant, and when evidence about its plans and actual behaviour demonstrate that the agent is 'in control' of its own behaviour, we may trust that the agent will indeed behave in a compliant way.
– Finally, for value-based architectures that allow for ethical decision making, we may also use evidence of the social values of the agent, and how they are effective in actual decision making. However, the causal influence of stated values on actual decisions is hard to demonstrate. Therefore, such compliance is similar to the establishment of trust. Similar to identification-based trust [20], identification of the values of an agent may be based on cultural 'cues'.

Regarding the debate between rule-based versus principle-based norms, principle-based systems are much harder to implement and monitor. Principle-based systems require that subjects and auditors have a deliberative agent architecture, and often also the possibility to deal with trade-offs by ethical decision making. This could explain why principle-based norm systems may degenerate into rule-based systems.

Regarding the use of sanctions, we can say that sanctions assume agents are both willing and able to adjust their behaviour. This means that sanctions must really be undesired; also internal sanctions (guilt) may work. When agents lack expertise to improve their behaviour, giving advice may be more effective than sanctions.

The analysis has been validated in a case study of horizontal monitoring, a new auditing approach established by Dutch Tax. This auditing approach is based on mutual trust and understanding. It assumes companies have an interest in being compliant. Traditional command and control often works counterproductive. Instead of hunting mistakes in tax declarations, auditors must now give advice to companies on how to avoid misstatements. However, because it is principle-based, horizontal monitoring assumes that companies have an internal decision making structure which is at least goal-based and preferably also allows for ethical decision making. The case study shows that, in addition to evidence of behaviour, horizontal monitoring also requires evidence that compliance issues are taken into account during internal decision making.

References

1. K. Atkinson, T. Bench-Capon, and P. McBurney. Computational representation of practical argument. *Synthese*, 152(2):157–206, 2006.
2. I. Ayres and J. Braithwaite. *Responsive Regulation: Transcending the Deregulation Debate*. Oxford University press, 1992.
3. G. Boella, H. Verhagen, and L. van der Torre. Introduction to the special issue on normative multiagent systems. *Journal of Autonomous Agents and Multi Agent Systems*, 17(1):1–10, 2008.
4. V. Braithwaite. Responsive regulation and taxation: Introduction. *Law and Policy*, 29(1):3–10, 2007.
5. J. Broersen, M. Dastani, J. Hulstijn, and L. Van der Torre. Goal generation in the BOID architecture. *Cognitive Science Quarterly*, 2(3-4):431–450, 2002.
6. R. Conte and C. Castelfranchi. *Cognitive and Social Action*. UCL Press, London, 1995.

7. C. Castelfranchi. Prescribed mental attitudes in goal-adoption and norm-adoption. *Artificial Intelligence and Law*, 7(1):37–50, 1999.
8. R. Conte and C. Castelfranchi. From conventions to prescriptions: towards an integrated view of norms. *Artificial Intelligence and Law*, 7(4):323–340, 1999.
9. R. Conte, C. Castelfranchi, and F. Dignum. Autonomous norm acceptance. In *Intelligent Agents V: Agents Theories, Architectures, and Languages*, LNAI 1555, pages 99–112. Springer Verlag, 1999.
10. COSO. Internal controlintegrated framework. Technical report, Committee of Sponsoring Organizations of the Treadway Commission (COSO), 1992.
11. F. Dalpiaz, P. Giorgini, and J. Mylopoulos. An architecture for requirements-driven self-reconfiguration. In P. van Eck et al, editors, *Proceedings of CAiSE'09*, LNCS 5565, pages 246–260. Springer Verlag, 2009.
12. W. E. Deming. *Out of the Crisis*. MIT Center for Advanced Engineering Study, 1986.
13. F. Dignum. Autonomous agents with norms. *Artificial Intelligence and Law*, 7:69–79, 1999.
14. U. Gneezy and A. Rustichini. A fine is a price. *Journal of Legal Studies*, 29(1):1–17, 2000.
15. H. Gribnau. Soft law and taxation: The case of the Netherlands. *Legisprudence*, 1(3), 2008.
16. K. Hindriks and M. B. van Riemsdijk. Satisfying maintenance goals. In *Declarative Agent Languages and Technologies V*, LNAI 4897, pages 86–103. Springer-Verlag, 2008.
17. W. S. Humphrey. Characterizing the software process: A maturity framework. Technical Report CMU/SEI-87-TR-11, Carnegie Mellon University, 1987.
18. IAASB. Audit sampling: Redrafted international standard on auditing (ISA 530). International Federation of Accountants (IFAC), 2008.
19. L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.
20. R. J. Lewicki and B. B. Bunker. Developing and maintaining trust in work relationships. In *Trust in organizations*, pages 114–139. Sage Publications, Thousand Oaks, CA, 1996.
21. R. Lu, S. Sadiq, and G. Governatori. Measurement of compliance distance in business work practice. *Information Systems Management*, 25(4):344–355, 2009.
22. P. Maes. Concepts and experiments in computational reflection. In *Proceedings OOPSLA*, pages 147–155, 1987.
23. F. Meneguzzi and M. Luck. Norm-based behaviour modification in BDI agents. In *Proceedings of AAMAS'09*, pages 177–184, 2009.
24. M. E. Pollack. Plans as complex mental attitudes. In P. Cohen, J. Morgan, and M. Pollack, editors, *Intentions in Communication*, pages 77–103. MIT Press, Cambridge, Mass., 1990.
25. A. S. Rao and M. P. Georgeff. Modeling rational agents within a BDI-architecture. In J. Allen et al, editors, *Proceedings of KR'91*, pages 473–484, 1991.
26. S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, New York, 2nd edition, 2003.
27. Sarbanes and Oxley. Sarbanes-oxley act of 2002. Public Law 107 - 204, Senate and House of Representatives of the United States of America, 2002.
28. D. Satava, C. Caldwell, and L. Richards. Ethics and the auditing culture: Rethinking the foundation of accounting and auditing. *Journal of Business Ethics*, 64:271-284, 2006.
29. M. C. Schut. *Scientific Handbook for Simulation of Collective Intelligence, version 2*. Available under Creative Commons License, www.sci-sci.org, 2007.
30. J. Searle. *The Construction of Social Reality*. The Free Press, New York, 1995.
31. C. Sierra. Agent-mediated electronic commerce. *Journal Autonomous Agents and Multi-Agent Systems*, 9(3):285–301, 2004.
32. R. Thomason. Desires and defaults: A framework for planning with inferred goals. In A.G. Cohn et al, editors, *Proceedings of the International Workshop on Knowledge Representation (KR'00)*, pages 702–713. Morgan Kaufmann, San Mateo, CA, 2000.
33. E. Visser. Tax control framework. Dutch Tax and Customs Administration, 2008.