An Agent-based Legal Knowledge Acquisition Methodology for Agile Public Administration

Alexander Boer
Leibniz Center for Law
University of Amsterdam
Amsterdam, the Netherlands
A.W.F.Boer@uva.nl

Tom van Engers
Leibniz Center for Law
University of Amsterdam
Amsterdam, the Netherlands
T.M.VanEngers@uva.nl

ABSTRACT

This paper proposes a knowledge elicitation method based on serious gaming for theory construction about the effects of the law on the behaviours of agents. These games provide input to simulations of business process and product design alternatives. For knowledge representation, we have combined agent role descriptions with a generic task framework. An important thesis of this paper is that, in the interest of quick and simple domain analysis, agent roles, not intelligent agents, should be the focal object of simulation of complex social organizations. At least if getting a grip on social complexity is the purpose of modeling.

Keywords
Legal Knowledge Engineering, Public Administration, Knowledge Acquisition

1. INTRODUCTION

At least some large organizations in public administration, like tax administrations and immigration services, can be characterized as subject to almost continuous change of the law. Many reorganizations with different deadlines happen in parallel, implementation often requires coordination with network partners (like employers, schools, accountancy firms, banks and insurers, etc), and anticipated legislation is sometimes even delayed because responsible organizations cannot keep up. Early delivery of workable process and product specifications for public administration, as it implements new law, is a big challenge for developers in public administration, and failure causes backlogs, waste of money, litigation, and injustice.

In the Agile project [6, 5] we work with the Dutch tax administration and immigration service, to make legislative drafters, product developers, network partners, and people working in service delivery processes, work together on a daily basis, to share expert knowledge and to prioritize product and process development. To this end they need a knowledge acquisition methodology that is quick and effective, and helps them to focus on justifying design activity with reference to the law. This knowledge acquisition process for public administration is expected to leverage knowledge from:

1. the legal rules as evidenced by the sources of law, but
   also
2. design experiences and familiar design patterns used
   in the organization,
3. existing resources and social structures that should be
   preserved or adapted, and
4. experience or evidence-based expectations about the
   behaviour of agents in the relevant domain.

The organizations involved in the project are already committed to various methods of specifying business processes, law-based business rules, XML message protocols, domain conceptualizations, etc. The perceived problem is the disconnection between various modeling perspectives on the organization, and a resulting lack of agility of the organization. Attempts to agree on shared domain conceptualizations and standardized transactions with network partners are often perceived to contribute to this problem, rather than solving it.

The knowledge elicitation method we proposed for the Agile project is a form of serious gaming, aiming for theory construction about the behaviours expected of agents in the domain in reaction to the changes. An innovation over compliance frameworks leaning heavily on the sources of law, is that the methodology stresses the elicitation of noncompliance stories, to allow structured integration of experience of critical incidents [14] in design activities. We learned that specifications insufficiently leverage experience about the expected behaviour of agents in the domain. This weakness is most evident in the absence of explicit theory construction about noncompliance in design, even though experience about critical incidents in the past, and the ability to predict future incidents, is usually available in the organization and certainly plays a leading, but largely implicit, role in design.

In the associated knowledge representation framework the notion of agent roles, and the possibility of multi-agent system simulation, therefore play a central role. We argue in section 2 that the distinction between agents and agent roles is ill-advanced in this context: agent roles should be the focal object in simulation of complex social organizations. We believe this view has consequences for agent coordination in multi-agent systems.
In the interest of leveraging existing structures and design patterns in the organization, we merge the agent role concept with an existing generic task framework in section 3. This framework allows for robust generalized business process designs, and explains certain knowledge level design patterns, not addressed by other business process specifications already in use, that we encounter in organizations. We extended the generic task framework with three abstraction levels to accommodate our view of the design and legislation processes, and the motivating role of critical incident stories from the work floor in these processes.

In section 4 of this paper we discuss some aspects of an example of an agent role description. The larger context of the knowledge acquisition game is addressed briefly at the end, in section 5. Finally, in the discussion in section 6 we discuss some interesting relations with existing research, and summarize conclusions and ways to move forward.

2. AGENT ROLE MODELING

Agent roles in general are identified with a set of abilities, and a set of susceptibilities to actions of others, and associated with goals, plans, and beliefs typical of that role [23, 20, 4]. The purpose of gaming as a knowledge acquisition approach is elicitation of agent role descriptions from experts, as opposed to the development of effective agent player strategy when agent roles are already well-understood.

Law-based agent roles are a natural approach to knowledge acquisition from sources of law. Law-based agent roles can be characterized by a set of powers and liabilities, duties and rights in legal institutions conceptualized by the lawmaker [4, 3]. To adopt or attribute a law-based role is to adopt the conceptualization as given by the law. The relationship between rules, institutions, and the analysis of agent interaction in terms of powers and liabilities, duties and rights is linked to an important tradition in our field that analyzes Hohfeld’s legal relationships between people [24].

Agent roles are played by an agent player. The relation between roles and players can be very complex: agents can play multiple roles, a role can be played by multiple agents, roles can play roles, and a role can survive the agent that plays it [3]. In the law we typically find the full range of complexity in the relationships between players and roles. We propose that a lot of the complexity in legal reasoning is found in the assignment of agent roles to players, and coordination between the demands of agent roles. The complexity is there, giving rise to problems and opportunities, regardless of whether this coordination takes place inside the mind of one player who adopts multiple agent roles, or multiple players who communicate to coordinate.

In the next subsection we position this claim in the context of multi-agent systems (MAS) research, postulating that the agent in MAS should be understood as the (instantiated) agent role rather than the player. In subsection 2.2 we briefly address the individuation question that arises when one makes this claim about agent roles.

2.1 Agent roles and multi-agent simulation

Multi-agent system (MAS) simulation in Agile is mainly intended as an aid for understanding the domain and the relevant agent roles in it, for animation of interesting stories, and for early prototyping of design solutions. In this section we discuss the way in which we use the MAS paradigm.

The classic multi-agent system (MAS) paradigm is based on the assumption that, in principle, any agent may communicate with any other agent it knows [13]. It is therefore the responsibility of the individual agent to constrain access to the services it provides and to contribute to greater organizational control objectives. The paradigm in its basic form proposes internalization of agent coordination solutions as knowledge of the agents, instead of externalization in the form of structures outside the agents that constrain and guide interaction. A key role is played by the agent’s ability to identify other agents that it may interact with in certain ways [13].

Many extensions to the MAS paradigm have been proposed to accommodate and standardize complex social organization, often with MAS-based simulation of real world organization and coordination problems in mind. Proposed extensions add a framework for describing organizations, or institutions, or agent roles, or normative systems, or some combination of these concepts [12, 28, 13]. These extensions are often externalized, in the sense that agent behaviour is directly guided by organization artifacts that exist in the MAS environment outside the agents [13]. In the area of normative multi-agent systems (NMAS; [2]), compliance frameworks like e.g. [29] are externalized normative systems. Externalization of law and social organization is philosophically defensible, and often an efficient solution from a systems engineering point of view.

For simulation purposes, externalization is however often not an elegant solution, regardless of whether you think that the organization or the legal rule is really out there. When you propose the MAS environment as a knowledge representation, which is what you do if you propose MAS-based simulation, its externalized aspects become common knowledge of something else being simulated. If the purpose of simulation is understanding the domain from the relevant points of view, it does not prima facie make great sense to treat the aspects one tries to understand as common knowledge. Implicit reliance on common knowledge structures leads to overlooking information sharing processes in organizations that are in reality less than perfectly objective and reliable.

Another central assumption in MAS, regardless of whether one uses internalization or externalization as a solution for coordinated action and complex social organization, is that the MAS agent is a player, the one who really acts, while the social role the agent is acting in, is an internal or external artifact. In theory construction about law with domain experts, it is however more natural to think of the agent player as an internalized artifact, optionally present, of arbitrary complexity, of the acting agent role. Agent simulation is, if we adopt this view, primarily simulation of interaction between persistent and reified agent roles, and agent players are only allocated to agent roles if there are specific relevant stories that depend on a notion of agent player identity across agent roles. There are several theoretical and practical reasons to prefer this approach.

Firstly, the distinction between agent roles and agent players is found in the distinction between subjective descriptions of institutional or social realities, and an objective brute reality that subjective realities supervene on [18], in which real players of flesh and blood exist. This distinction is influential but philosophically tenuous [18]. More often we find that agent roles in some institutional context play other agent roles in another institutional context. Clearly, in law,
reduction of agent identity to natural persons is practiced in some contexts (e.g. crime) and not practiced in other contexts (e.g. commercial law). In public administration practice, flexible theory construction about agent roles and players is of great importance. Awareness of player identity of the role player is key strategic information in many non-compliance scenarios, and not common knowledge. Description and simulation of such scenarios requires that player identity is a subject of belief.

For instance, civil law constrains the behaviour of the buyer and seller. If we want to simulate the possible behaviours of buyers and sellers, these are properly modeled as agent roles. Common sense dictates that normally seller and buyer agents are played by different players, but there are in fact a number of tax evasion schemes that assume that an agent knows that he is selling to himself in some other role, and that the tax administration is unaware of this fact. For the tax administration, the seller who sells to himself, and tries to misrepresent that fact, is an agent role abstracted from past experience. This abstract agent role generally survives changes in tax law, and may be brought up in discussions with domain experts.

There is, additionally, general reason to localize attributed beliefs, goals, and plans to agent roles instead of agent players. The law sometimes depends on the assumption that agents are able to 1) act in a specific assigned role with privileged access to information without, for instance, compromising confidentiality and privacy, 2) use abilities that they have in an agent role to reach the goals specific to that role only, or 3) commit to contradictory beliefs in different roles. To correctly model the intended influence of the law on agent behaviour, including the complex internal logic of noncompliance stories, we attribute beliefs, goals, and plans to agent roles.

For instance, the issue with sellers who sell to themselves is not the identity issue as such, but the apparently uneconomic price set for the property exchanged. The seller has the wrong goal – tax evasion – in mind when he determines the price of the property. This places the seller that sells to himself scheme in a broader category that includes kickback and extortion schemes, all characterized by a price deviating from the reference market value, from the tax administration’s point of view. The nature of the relationship between agents, and the direction of the deviation from the reference market value determines the hypothetical scenarios a tax administration may consider.

Lastly, the acting agent role allows us to address the question of internal vs. external coordination of agent roles in a transparent way. If agent role A for instance informs agent role B of something, it usually does not make a substantial difference whether A and B are played by the same agent player P or not. When both roles are played by the same player, the communication event is internal to that player, while otherwise it is external. In both cases, a third agent role C does not become aware of the communication if it is not included in the loop by either A or B. The common sense idea that the agent player in principle has access to all beliefs it holds in the various roles it plays can be seen as an attributed goal (and ability) of agent roles played by the same player to share beliefs that may or may not be present.

In our seller example, the buyer and seller have determined a property price that deviates by some specific amount, that is to be transferred between buyer and seller for some reason, from a reference market value. Whether this happened inside one mind, or between natural persons, is relatively immaterial from the tax administration’s point of view, since it is not a witness of this price determination anyway.

Explicit externalized communication of information between agent roles considerably simplifies flexibility of allocation of agent players to agent roles because information access patterns, even when they are conceptually speaking internal, are explicitly modeled at an early stage of design. This significantly contributes to the agility of the organization in resource allocation and business process design.

To summarize, the MAS paradigm is most useful if we understand the agent to be a simple, one-dimensional agent role, with player identity and other coordination artifacts internalized as beliefs. While any simulation can be achieved with agent roles as artifacts used by MAS agents, this firstly adds an often unnecessary layer of complexity to MAS, and secondly forces us to distinguish between internal and external solutions to essentially the same agent role coordination problems in modeling. For these reasons we prefer to simulate agent roles instead of agent players as MAS agents.

### 2.2 Selecting agent roles

An important design question if we decide to focus on agent roles rather than agent players, is how one selects appropriate agent roles. Which agent roles exists? Why do we distinguish or aggregate agent roles? We distinguish three basic types of agent role: 1) law-based, 2) functional, and 3) experiential agent roles.

Differentiation of law-based agent roles is a matter of deciding whether an agent term, for instance seller, found in a unit of discourse in the law addresses the same agent role as usages of that term in other units of discourse in the law. This is an issue in modeling of the law in general that we addressed earlier (see ch. 5 of [4]). Although it is an interesting subject, we do not address it here.

Functional agent roles are created by the organization, and represent an internally oriented design perspective on the organization, on the organization as a system of which the agent role is a functional component. In both the functional and law-based agent role description, expectations are of a prescriptive nature. That does not mean that law-based agent roles (e.g. thief) are always functional from a design perspective.

Experiential agent roles represent an external and descriptive perspective on the organization, on the organization as an an agent in an environment that includes clients, network partners, and employees, whose behaviour is understood in terms of experience. This type notably includes risk profiles used to select potential cases of noncompliance for monitoring. Note that noncompliance risk profiles are equally applicable to employees, network partners, and clients.

Pragmatically, aggregation and subsumption of roles are leaving aside the operational semantics of these notions in a specific MAS environment – mainly a question of interchangeability in a system simulation. Important to recognize is that the part replacement logic for law-based and functional agent roles supports the generation of design hypotheses, while the replacement logic for experiential agent roles supports hypothesis-driven monitoring and diagnosis of a design. To understand the purpose of both types of agent roles, we need to understand them in the context of
a organization design methodology. In the next section we present a generic task methodology for public administration based on a generalized problem solving framework that covers design and diagnosis, creating a context of design patterns for the agent role descriptions.

3. GENERIC TASKS IN PUBLIC ADMINISTRATION

Design activities in public administration are obviously not based on interpretation of the law alone. The organization has invested in existing structures and resources, and therefore tries to fit new law-based tasks into existing structures. It is therefore not surprising to find some notion of generic tasks or service design patterns in organizations, that are replicated in different places with different legal content.

We will first present a typical design pattern to familiarize the reader with this notion, and then move on to the framework for generic tasks we use and the associated knowledge acquisition method.

Fig. 1 shows a typical service delivery process description as we often find it in public administration. We have chosen a very general level of description for the tasks, and do not commit to a specific workflow coordination option with explicit choice points, etc. The figure shows three main business process types (large squares) in a common configuration: service delivery (top), legal assessment (bottom left), and enforcement (bottom right).

Even on this level of generalization, workflow coordination is of course not entirely arbitrary from a legal point of view. In some cases the organization is for instance obliged to collect evidence or obliged to enforce, and this limits the possible workflow coordination options. Compliance frameworks characterized by an externalized normative system, e.g. [22, 29], excel in this aspect of design compliance. We believe this is a less pressing and better understood knowledge management problem in public administration: design compliance of this sort does not address robustness and agility. In business, where law is less central to operations, compliance frameworks do appear to address an urgent problem.

In Fig. 1 we have, rather arbitrarily, chosen for a basic distribution of tasks over agent roles D for request handler, H for request handling, M for monitor, and L for the law department desk. Any other distribution

is possible, and many others are at least equally reasonable. Another arbitrary choice is to assume that third party contacts take place in client request handling, and that no client contacts take place in risk assessment, etc.

Less arbitrary are two business process design patterns we want to draw attention to. The progression of classification, planning, and execution with external information gathering contacts is an invariant structure in service delivery processes. This progression is repeated three times in Fig. 1. It represents a generalized problem solving paradigm.

The second common invariant structure is the background diagnostic process behind any service delivery process that fits available evidence to noncompliance stories, and the enforcement process that initiates enforcement or litigation. Diagnostic and enforcement processes may be run routinely or opportunistically, depending on a cost-benefit analysis. Whether third party information is used, or additional information from the client is solicited, also depends on a cost-benefit analysis and perhaps on whether duties to inform exist at the side of the client.

These monitoring processes typically consume a lot of organizational resources in comparison with primary service delivery processes, but they are often less well described from a knowledge management point of view. The two design patterns fit well to many tax administration, immigration, and other public administration services.

A simple example: A client may request a residence permit on the grounds of being married to a resident of the country. In the primary service delivery process a marriage certificate will be asked, and the validity and currency of a certificate may be verified with the third party that registered the marriage, before a decision is taken. The organization may however also decide to schedule a monitoring process to verify that the marriage does not fit the stories for a sham marriage, in which case it may enforce. The time frame for this monitoring process, and its objectives, completely dissociate it from the service delivery process.

On a more generic level the design patterns also work well for the sales example used earlier. We can for instance replace the announcement of the decision with delivery of the good sold, and the diagnostic process now concerns whether the obliged payment takes place. Enforcement processes vary from sending a friendly reminder to filing a petition for bankruptcy to recoup the money.

We can also turn things around if we want, addressing the other side of obligations, leading to another important variant of the diagnostic design pattern: in other prototypical sales processes the payment was made in process by the client, and the obligation to deliver is the noncompliance risk to monitor for. The organization may set itself the task to deliver, but may fail to timely do so. In this case a remedial action to prevent noncompliance, litigation, or enforcement by the other party is asked for, instead of litigation or enforcement against the other party.

Functionally similar monitoring processes in public administration monitor decision deadlines, the legality of use of legal powers, compliance with privacy-related information access restrictions, etc.

The generic problem-based design patterns assist us in attending to the functionally different internalizations of a legal rule, for instance an obligation, in different tasks and different agent roles, and differentiates the knowledge contexts in which the legal rule is applied.
The opportunistic aspect of many real world compliance monitoring problems focuses our attention on noncompliance stories and evidence trails, value and cost of information, the use of cheap indirect information to preselect cases for compliance monitoring, the use of statistical control groups and the costs of using them, and theory construction about the relevance of missing information [25]. Very technical developments in administrative case law can often be understood from this point of view. The alternative externalized norms approach to compliance trivializes the detection of violations and does not address these technicalities.

We present a functional classification of generalized task design patterns in public administration in Fig. 2 (used earlier by us in [6, 5]). Its theoretical basis in knowledge engineering theory is discussed in the next section.

### 3.1 A generic problem solving paradigm

Our typology of generic tasks in public administration is based on the typology of problems and views on problem solving presented two decades ago in [7]. This typology, the *suite of problem types*, was based on an analysis of the problem and task decompositions found in knowledge-based system literature [10, 11, 26]. Important theses of [7] were (amongst others) that 1) the availability of structural and behavioural models in a domain determines which problems can be posed and solved, 2) there are recurrent functional dependencies between problem types, 3) tasks package chains of dependent types of problems, and 4) reusable problem solving methods match to tasks.

The *generic problem types* distinguished in that paper – for instance design, planning, scheduling, monitoring, assessment, diagnosis, etc. – do a good job in our experience of describing recurrent design patterns found in knowledge-based system components found in public administration.

The proposed general approach to knowledge-based systems analysis consists of three phases: moving from identification of an ill-defined problem to (well-defined) problem definitions, each with a problem space and an abstract solution, and then to task specifications. In public administration, the change in the law often plays the role of an ill-defined problem.

The suite of problem types presents us with a generic problem solving cycle, and two different vocabularies for describing it, depending on the type of model of the domain that is available. For the purposes of this paper we have condensed the suite of problem types into two chains, representing two alternative perspectives on what is in essence the same generic problem solving cycle.

1. **Model → Design → Implement → Monitor → Diagnose**
2. **Classify case → Plan → Execute → Monitor → Diagnose**

When we feel able to control a domain by encapsulating processes into fixed structures we decide on a design for dealing with a type of problem and implement it. When we feel this is not feasible, given the characteristics of the domain or problem, we make a situation-specific plan to address a problem. A fully articulated problem solving cycle would in principle allow description in both terminologies, but functioning systems generally address some problem as planning problems and others as design problems.

The opportunistic aspect of many real world compliance monitoring problems focuses our attention on noncompliance stories and evidence trails, value and cost of information, the use of cheap indirect information to preselect cases for compliance monitoring, the use of statistical control groups and the costs of using them, and theory construction about the relevance of missing information [25]. Very technical developments in administrative case law can often be understood from this point of view. The alternative externalized norms approach to compliance trivializes the detection of violations and does not address these technicalities.

We present a functional classification of generalized task design patterns in public administration in Fig. 2 (used earlier by us in [6, 5]). Its theoretical basis in knowledge engineering theory is discussed in the next section.

Part of the problem solving activities may of course be delegated to the human user of the system. Planning/scheduling systems without appropriate monitoring and diagnosis functions to detect important failure modes are for instance quite common in practice; Replanning if unexpected things occur is left to the human user.

We proposed a classification of types of monitoring and diagnosis processes in [5] which will not be repeated here, distinguishing the following three basic types:

1. **regulative function monitoring**
2. **constitutive function monitoring**
3. **performance monitoring**

These three types of processes monitor events based on three different domain conceptualizations: the organization and its environment as a normative system, as an ontologically coherent institution, and as a goal-directed agent in an environment. In the context of monitoring experiential agent profiles are used as diagnostic hypotheses.

The sales example provides us with examples of each type: Monitoring whether the seller delivers and buyer pays is regulative. Monitoring whether the verbal agreement to a real estate sale is effectuated in a written contract\(^3\) is constitutive. Monitoring whether sales take place at acceptable price points is usually for performance reasons, although it can be equally relevant as compliance datum, as noted before.

Fig. 2 also expresses an additional observation we made, beyond the suite of problem types presented in [7]. We distinguish three abstraction levels of problem solving, and propose a generic description of how these three levels are related. Abstraction levels in reasoning were hinted at but not articulated in [7]. In public administration it is obvious to think of three clearly differentiated levels. At each level actions need to be recorded and justified with reference to the law.

At the basic level individual case handling level plan switching takes place based on case classification, which is a trivial approach to the modeling problem. This is in our view the

---

\(^3\)This is a requirement in Dutch law, in exception to normal rules for sales which only require some expression of offer and acceptance.
reactive adoption of predefined agent roles based on communication events like the client service request that the agent is reactive to. In the example business process design discussed earlier the service delivery process (Classify case → Plan → Execute), noncompliance monitoring (Monitor → Diagnose), and enforcement or litigation (a re-entry to (Classify case → Plan → Execute → Monitor → Diagnose) take place at this first level.

At a higher level creative design and resource scheduling takes place, mainly directed to the refinement of agent role descriptions and other knowledge sources used, and the systems and structures supporting agent activity used at the basic level. Design activity is mainly a result of outcomes of diagnostic processes in individual case handling (both primary service delivery and litigation diagnostic information) translated into requests for reallocation of resources or change of supporting knowledge sources, systems, and structures on the design level.

When design level problems cannot be solved within the constraints of existing policy, diagnostic information will be forwarded to the diagnostic processes on the policy making level, leading to proposals for new policy. This is usually an informal line of communication (in the sense of usually being outside the scope of explicit communication models) from a committee of domain experts responsible for analysis of legal problems to the competent legislative drafting department and the responsible minister that has to sponsor the proposal.

Fig. 2 reflects our interpretation of design and policy making as core activities of the organization. Public administration and policy makers are mutually dependent; Public administration is in charge of efficient operationalization, and the policy maker wants effective rules. Process and product development is a core activity in public administration that consumes a large and increasing part of its resources. Overspending on IT projects and reorganization, and slow policy making processes attract increasing attention. This means that knowledge management, process and product development, and resource allocation for the process and product development process is an increasingly pressing issue. Initially attention will go to work flow and reliability and reusability of knowledge representation, but there is a potential for monitor-based automated resource scheduling and business process realignments.

4. AN AGENT ROLE EXAMPLE

The approach to MAS simulation of agent roles in the Agile project [6, 5] can be explained in the form of a simple example. The key question to address is how the generic task design patterns help to adequately describe an agent role, and how generic tasks are represented as knowledge imputed to agent roles.

We assume that the reader of this paper understands the process of buying, and knows of different procedural variants of this process. There is a sale when there is an offer and an acceptance of that offer, regardless of buyer or seller initiative, and the sale leads to a duty of the buyer to timely pay the price to the seller, and a duty of the seller to timely deliver the property to the buyer. There are also exceptional requirements to this description of the transaction. Real estate sales agreements must for instance be written down to count as a sale, and lead to a duty to register the transaction in the Cadastral registration. There are also rules for remedies if duties can not be complied with. In this section we ignore the identity of buyers, property, price, time, and many other things that would make the example more interesting and more complicated.

Let us assume we are designing the business process of a seller. The presentation of generic tasks suggests we have to distinguish service delivery planning on the one hand, and diagnosis tasks on the other hand. An organization has to attend to the following goal/event-condition-action statements to describe the service of selling:

1. If the seller has an interest in a sale, then the seller has the intention to offer, and then to monitor whether an acceptance and a sale takes place;
2. If an offer is made to the seller, and the seller has an interest in a sale, then the seller has the intention to accept, and to monitor whether a sale takes place;
3. If the seller believes he made a sale, then the seller has the intention to monitor payment; and
4. If the seller believes he made a sale, then the seller has the intention to deliver.

Translating this into a routine work flow involves expectations about the buyer. If buyer for instance requires delivery before paying, and seller requires payment before delivery, no successful sale can come about. When one of the parties is a natural person, organizations expect that they will show the required flexibility to conform to the procedure the organization decides on, but it is a fact that some organizations will have trouble buying from some sellers because bureaucratic procedures are incompatible.

To separate work flow and decision making from the abstract agent role description, we distinguish in knowledge acquisition between intentional states \( int(P) \), representing commitment to bringing about an event in \( P \). \( mon(P) \) to represent a commitment to monitoring \( P \), and \( do(P) \) to represent commitment to bringing about \( P \). When the agent decides to perform a task \( do(P) \), it usually has the intention to monitor performance \( mon(P) \). These are purely ontological conventions: the operators have no special semantics in MAS execution until additional work flow decisions are taken.

The agent role description from law is a few design decisions short of a simple, executable agent role. Consider the Jason AgentSpeak plans in Fig. 3, that is a simplified implementation of an abstract seller, with \( S \), \( O \), \( P \), \( D \) being sale, offer, payment, and delivery. \( S + B \) means that \( B \) becomes a goal if a belief (or a goal) \( B \) is added, and \( + \) is a sequence operator. Besides the action goal \( !G \), there are test
goals ?G. In abstract agent role descriptions only do and mon action goals are adopted. In the example, belief only plays a role as condition to the payment and delivery obligations (+S or now there is a sale). There is no worked out plan for adopting such a belief. Work flow decisions limit the number of different possible executable implementations:

1. the agent always performs plan steps in a specific order; and
2. the agent waits until it believes that a plan step has been successfully completed before it starts another one.

The example agent role description commits to initiating payment monitoring before initiating delivery. This is a design decision beyond the text of the law. This is however not the same thing as committing to completion of monitoring of payment before delivery, which would be (trivially) achieved by the statement that a belief in payment should be adopted before mon(P) completion:

+!mon(P(S)) <- ?P(S). //wait for belief

This is a simplistic payment monitor implementation, and completely embeds the monitoring process part of the service delivery work flow.

It is possible to program an agent to dynamically schedule task execution in AgentSpeak. We do not assume this ability, because execution forces one to take work flow decisions, and flexibility in scheduling is obviously also a work flow decision that is often not desirable.

 Normally speaking, the monitoring process for the sale, payment obligation, and delivery obligation will be layered in a large organization. In service delivery planning verification of a payment order or bank transfer may for instance meet the proof standard for payment, and allow the plan to move forward to completion, but there are, even in this simple example, a number of long term threats to the achievement of a sale:

1. the payment order is cancelled or the payment is rolled back (storno);
2. the delivery is contested by the buyer, because delivery of the good did not take place or the good does not meet expectations; or
3. the sale is contested (e.g. parent contests legality of sale to a minor).

These exceptions can be treated as a default reasoning problem for an integrated seller agent, but will in a large organization often be delegated to other agents. These events are rare and diagnosis and reaction require special knowledge and abilities not trusted to the average seller (e.g. handling a storno), may require interactions with other agents, and may naturally interact with other tasks only tangentially related to the seller (e.g. bookkeeping, procurement, maintenance, etc.). The seller therefore, in addition to monitoring with a relaxed proof standard in the transaction, delegates monitoring tasks to specialized diagnostic agents:

+!mon(P(S)) <- .send(payment-monitor, achieve, mon(P(S))); ?P(S).

As suggested by Fig. 2, the diagnosis by the payment-monitor may lead to feedback to service delivery planning by the seller; in the case of an unjustified cancellation of payment or storno, the diagnostic agent may for instance exclude an unreliable buyer from future transactions with the seller. This can be implemented in various ways, depending on the way the topological connection between buyer and seller comes about, not addressed in this paper. The payment monitor addresses constitutive function (the relation between a proof of payment order and payment), regulative function (reaction to the buyer’s violation of the duty to pay), and performance function (by excluding known unreliable buyers from transactions with the seller agent role) in relation to payment.

The diagnosis problem of the payment-monitor can be conceived of as the attribution of an experiential agent role profile to the buyer. The utility of this approach is not immediately evident with only a single buyer as a one component system to be analyzed. In the more relevant real estate transaction example of section 2.1, the various tax evasion systems involve chains of buyers and sellers of varying length, notary lawyers, assessors, and real estate appraisers.

The generalized service delivery planning and monitoring design patterns covers the processes found in practice, and allows for alternative perspectives on the same legal rules, with different proof standards. The complexity that inevitably arises when one implements a MAS simulation, shows up in coordination and message exchange between agent role descriptions, instead of in one agent description that delivers services and deals with exceptional scenarios through default reasoning. The messages sent and the proof standards applied are for us reminiscent of some theories of argumentation in law (e.g. [15]), and we expect that externalization of coordination between agent roles is a first step off an alternative route towards a generic theory of proof standards that applies to both internal default reasoning processes and external information exchange processes [16].

The abstract seller description in Fig. 3 minimally commits to a belief update strategy, minimally restricts work flow, and does not tell us who executes the various tasks. The description describes an abstract seller that is almost executable, but without a need for embedding domain knowledge about the acceptability of prices, etc.

Experience with knowledge acquisition games teaches us that the description in Fig. 3 is also representative of the level of detail experts will choose in playing out a story, with an hour of gaming translating into less than two pages of agent role description.

5. KNOWLEDGE ACQUISITION GAMES

The framework we sketched so far naturally lends itself to the use of structured knowledge acquisition games. In the Agile project we have tested such a game with domain experts from a participating organization. We briefly describe the game framework here, and defer discussion to section 6.1.

Gaming sessions are played with several to several dozen players, involved in 1) process and product design, and optionally 2) primary business processes, for instance if transitioning from existing infrastructure to a new one is an important aspect of the change process, and 3) policy making, for instance if an object of the game is to provide feedback on policy.
The three layer structure of figure 2 is the basis for definition of player role types. In addition, the planning and execution of service delivery plans and monitoring plans is distributed to different players. Agent roles are assigned for process and product design activity, and - if feedback to the legislative drafters is an object of the implementation process - to the legislative drafting department. A Nomic-like quality [27] can be given to the game by making the complete group of players represent the policy maker. Nomic itself is of course a fine example of a serious design game.

All participants have access to the relevant electronic legislation, and the tools to efficiently make references to it in messages and private notes. Information exchange between players takes place electronically, through short messages, and is logged. A minimal electronic infrastructure consists of sending eachother emails with a carbon copy to a logging email address, but we are in the process of defining a generic MAS framework in which human players can participate as agents.

The logs are used by product and process development to gain insight into possible process decompositions and the information contents of messages. These form the input to a development cycle that focuses on gaining quick wins in productivity in the next serious gaming session. Over the course of serious gaming sessions and short development cycles, the rules of the game, and the environment it is played in, become increasingly formalized, and agent roles are automated in the MAS environment. The 1) agent roles and associated legal abilities and susceptibilities, 2) reasons for exercising a legal ability, 3) ways in which abilities may be exercised, and 4) what counts as evidence for purposes of exercise of legal abilities in service delivery and monitoring, become clear. Participants increasingly agree on the information state of the game at each point in time as they gain insight into the essentials of information exchange, and information exchange becomes a routine matter.

Learning objectives are directed to understanding the law and its implementation in the organization. The game objectives - the objectives that participants are trying to win - of each serious gaming session are of course distinguished from the associated learning objective. The choice of game is itself an essential part of the development cycle that takes place between sessions.

Game objectives are tailored to (re)creation of critical incidents. Expert experiences of critical incidents and relevant case law can be converted into game scenarios, and new critical incident scenarios and coping strategies for those incidents are uncovered in the course of the game. Our classification of monitoring strategies gives us a broad classification of types of incident to recreate. Clients may for instance be instructed to claim benefits they are not entitled to or to delay or lock case handling processes, employees or network partners may be instructed to extract confidential information, etc.

In our experience, experts tend to present their understanding of a domain in the form of anecdotes about critical incidents. An early pen-and-paper test with serious games at the immigration service showed that the serious game format indeed brings up many critical incident stories, and a basic outline of a business process. We expect that a MAS-based knowledge acquisition shell helps experts to present such stories in the context of a knowledge framework, and to keep focus on developing one story for some time instead of retrieving a dozen similar ones from associative memory.

6. DISCUSSION

In this paper we addressed three subjects in the context of knowledge acquisition from law: agent roles in MAS simulation, generalized problem solving, and serious games as a knowledge acquisition technique. The rationale for bringing these together in the proposed way depends in part on trends in related research presented next. We finish with conclusions.

6.1 Related research

Serious games have some claimed benefits relevant for knowledge acquisition. We do not assume that serious gaming necessarily leads to a greater yield of knowledge, or greater motivation, from the game participants. The objective is to increase the effectiveness of knowledge acquisition from the perspective of product and process development teams. Serious game development is claimed to be an effective motivator for the development teams that design them [9].

More importantly, by deciding on a restrictive and formal protocol for knowledge elicitation, the experts are encouraged to present their domain expertise in a form that is relatively easy to convert into a knowledge resource by product and process development teams. There is some evidence to suggest that relatively unstructured direct methods like protocol analysis often do not work very well compared to structured but indirect methods like card sorting (for taxonomies). At least in part this is because knowledge engineers fail to integrate the knowledge on offer if it doesn’t fit into an already existing knowledge framework [8]. This quality of card sorting is presumably shared by structured, indirect knowledge acquisition games. The game and agents concept appears to be a suitable integrative framework for knowledge representation. The fecundity of game theory in science over the last two centuries testifies to that. In our field argumentation games for instance take a prominent position.

The generic problem solving framework of [7] that we used, has been applied in the field of law in the nineties, giving rise to the view that problems solving in law can be characterized ex ante as legal planning and ex post as legal assessment [30]. Most theory construction in law (including argumentation) takes the ex post perspective. The story also makes an entry there: [19] addressed story understanding in legal responsibility assignment as a matter of reconstructing causal links between postulated events. This research did not address knowledge acquisition for noncompliance scenarios, assuming that knowledge about compliance (plus sufficient world knowledge) suffices for understanding non-compliance. The associated ontology [20] included mental terminology, but the work never had an agent-based component.

In argumentation, there is a noticeable trend towards use of argumentation schemes about evidence and use of stories in recent research [1]. For both subjects, agent role modeling makes great sense; both the causal links between postulated events in a story, and for instance the trustworthiness of experts and witnesses providing evidence, depend on theory construction about the motives and beliefs of agents. In terms of the problem solving framework, argumentation of represents a typical ex post view of law.
Interestingly, real world implementation of noncompliance monitoring in large organizations seems mostly disconnected from legal theory. There is a general notion of compliance data sets, that can be used for detection of noncompliance risk. This data set is not given that follows from a model of compliant or noncompliant behaviour, but a dynamic thing that is constantly refined and tested. The selection of data is to a large extent opportunistic: data is mined because it is easy to obtain and has some relationship with a noncompliance story. Theory construction is driven by evaluations of the forensic value of evidence in courts, and focuses on statistical analysis, reject inference, and missing information [25, 17]. To construct arguments against data mining evidence one needs a story explaining why the assumptions underlying data mining are not met in a monitoring setting.

We are interested in the theory construction that takes place when experts hypothesize what is in the relevant data set, and we believe that agent role descriptions are the central elements in the stories that experts develop in the design phase. A constructive approach to noncompliance monitoring uses these stories to define the data collection activities that the organization should have an interest in.

Finally, there is a lot of good research going on on representation languages for modeling the beliefs, goals and plans of other agents in an agent context, usually deriving from game theoretic concepts or epistemic concepts (cf. generally [31]). Much of the current research at least nominally takes place in the context of MAS and NMAS, addressed in section 2.1. MAS concepts have been integrated with organization and task-oriented knowledge acquisition methodologies, and organization models have been added to MAS frameworks [12, 28]. NMAS is specifically suitable for modeling compliance.

The concept of the intelligent agent remains an obstacle, as it tends to lead to a focus on rational strategy in well-understood social problem domains. The real world organization, on the other hand, is not a well-understood context of action, but a hypothesis constantly refined. In knowledge acquisition we focus on documenting what stories and evidence trails of critical incidents tell us about beliefs, goals, and plans that can be attributed to agent roles. The ability of intelligent agents to reflect on other agents and to internalize coordination issues, is interesting, but agents are not simple enough to be useful components during design. The simpler components are the agent roles.

6.2 Conclusions

The framework proposed in this paper accounts for, and accommodates, the critical incident stories that drive business process design, and deviates from the practice of simply seeing noncompliance as the logical complement of compliance. The story and agent roles approach to business process design helps the large organization to explore data mining assumptions like the plausibility of data being missing (not) at random, etc., in monitoring and enforcement policy. In figure 2 the integration of case law also becomes a natural and obvious thing. Case law presents an argument and evidence for that argument. Underlying these is a story. The causal links in that story usually depend on motives and beliefs attributed to agent roles.

Revised agent role descriptions – experiential or design-based – lead to revised business processes. The possible results of revision can be animated through MAS simulation. We hope to see more ties developing between agent simulation and argumentation in the future, with the agent role, story, and evidence trail as the focal concepts.

An important open problem area is coordination. Essentially, the claim made in this paper is that coordination between the demands of different agent roles within the mind of a single agent is essentially the same problem as the coordination problem between different role players in an external organizational structure. Internalizing coordination issues into the black box that the agent in MAS is supposed to be, does not contribute to our understanding of internal coordination stories. Externalized coordination is addressed by communication protocols, by sharing access to databases, by controlled work flow, by the use of a dedicated coordinator who solves scheduling problems, etc.

We expect that the same solutions we use for the organization can be metaphorically projected into the agent, resulting in agent roles only as active components; isn’t metaphor the source of concepts for describing the mind [21]?

Finally, an interesting future step is to find a conceptualization of the notion of burden of proof that applies equally to internal coordination between agent roles and external coordination between agents. We naturally expect burdens to be assigned to agents based on attributed interests in propositions; in the dialogical setting this is a proponent and respondent [15]. But expectations about distribution of burdens are already there in our interpretation of a legal rule [16], because legal positions are assumed to be in the interest of agents dependent on attributed agent role. Sellers have an interest in payments. Buyers have an interest in deliveries. This leads to expectations in monological reasoning about the buyer and seller. These expectations are occasionally upset: real estate fraud is an example of use of legal rules beyond the expectations already embedded in the rules. A general theory addressing this complication could be of great value in the field of compliance.

7. ACKNOWLEDGEMENTS

This paper is a result of the Agile project. Agile is a Jacquard project funded by the Netherlands Organisation for Scientific Research (NWO), that involves, besides the Leibniz Center for Law, the Technical University of Delft, and commercial and public administration partners. We are grateful to the Dutch Immigration Service and the Dutch Tax and Customs Administration for their support, and for allowing us to test our theories using interesting cases.

8. REFERENCES


