

Public Agility and Change in a Network Environment

Tom van Engers and Alexander Boer

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

Abstract: *Preparing for change is increasingly core business for governmental organisations. The networked society and the increasing connectedness of governmental organisations have as much impact on the complexity of the change process as the complexities of the corpus of law. Change is not only driven by changes in the law; changes in the organisation's environment often create a need to redesign business processes, reallocate roles and responsibilities, and reorder tasks. Moreover, preparations for change are not limited to the internal processes and systems of these organisations. Propagation of changes to network partners and redesign of network arrangements can be an enormous challenge. In the AGILE project, we develop a design method, distributed service architecture, and supporting tools that enable organisations — administrative and otherwise — to orchestrate their law-based services in a networked environment. This paper explains the Agile approach and describes some of its key principles.*

Keywords: legislation, change, adaptability, legal service, process, (non)compliance

Acknowledgement: AGILE is a Jacquard project funded by the Netherlands Organisation for Scientific Research (NWO). In the AGILE project, The Leibniz Center for Law of the University of Amsterdam cooperates with the Technical University of Delft. The Dutch Immigration and Naturalization Service (IND) and two companies, O&I and Be Informed, provide matching effort to the project.

Civil servants and citizens are equally expected to have knowledge about the law applicable to their spheres of activity. New law has to be applied from the moment it comes into effect. Organisations, public and private, therefore need some form of legal rule governance.

Many public administration organisations already support their employees by making legislation better- accessible and taking notice of new or changed legislation in an efficient way, but rule governance is not limited to document management. Rule governance includes all activities that enable organisations to cope with change. Inability to adapt in timely fashion to changed law, if only by creating a temporary, manual procedure, can create substantial backlogs and lead to unlawful action on a massive scale. This is of course not only inconvenient for the people that filed applications for some benefit and are waiting for case decisions, but it also results in huge compliance risks for public administration.

Keeping operational procedures up to date (i.e. effective and compliant with the legal requirements) at any time is very time consuming as well as highly expensive. Employees need proper instructions every time new or changed legislation comes into effect, and the supporting (legal) information systems need to be adapted. It is common knowledge that the maintenance costs are the largest part of the total cost of ownership (TCO) of an ICT system. Moreover, staff require additional training if the impact of maintenance on such a system is high.

Administrative organisations responsible for the effectuation of regulations have a great degree of freedom in designing administrative processes as long as they stay within the boundaries given by the legislator. And that design freedom is necessary. The design of administrative processes is for instance constrained by requirements following from the need to detect and react to non-compliant behaviour. As we will show, agent-based models that represent the positions, intentions,

beliefs, desires etc. of the different stakeholders that are relevant to the situations or acts described in the legal rules, help the designers of administrative processes anticipate compliant and non-compliant behaviour. Creating an administrative process compliant to the law taking into account non-compliant behaviour is one thing, creating it in such a way that it allows administrations to adapt to changes in the sources of law or changed circumstances is another. A second constraining factor is that changes often have impact outside of the administration primarily touched by those changes. Both changes in the sources of law and changes in government policy can result in the need for an organisation to create or renegotiate network arrangements with other organisations. Those other organisations also have their own rules and policies.

This kind of organisational change does not only affect the amount and complexity of rules the employees within those organisations have to deal with; It may also have major impact on organisational structures. All organisations affected may have to redesign their business processes and services, perform a mapping on existing roles and responsibilities, rearrange the order in which tasks are performed, and define new tasks. These kinds of changes may, over time, make the structure of an organisation more complex. Moreover, the rules the employees have to deal with grow in number and become more complicated too, because of interaction with other rules.

For these reasons rule governance, and specifically the change management aspect of it, are extremely important instruments for bureaucratic organisations that want to control their major asset, being their operational knowledge. This is equally important for non-governmental and governmental organisations. In this paper we will mainly focus on rule governance from the perspective of public administration organisations. Others are equally, but less fundamentally, affected by changes in the law, and have a similar need for a proper form of rule governance.

Arguably the ability of administrations to manage change in a clear, transparent, and cost effective way is crucial to the effectiveness of democratic states. Citizens have a right to expect that regulations that form the output of a democratic process are swiftly implemented without causing needless costs and implementation problems for those subjected to these regulations. Besides providing handles on a complex problem to the people working in public administration, the theoretical model that we will shortly present also provides better handles to those that want to comment on the legal system and its implementation, or want to bring about controlled changes to it.

This paper is a result of the AGILE project that specifically addresses the issues mentioned above. AGILE is an acronym for Advanced Governance of Information services through Legal Engineering. Within this project we aim at developing a design method, distributed service architecture and supporting tools that enable organisations, administrative and otherwise, to orchestrate their legal information services in a networked environment. The AGILE project started in the second half of 2008 and will last for four years. In this project we aim at improving the adaptability of ICT infrastructure, of business processes, and of data and knowledge within the organisation, given changing legal demands and constraints.

As stated, changes can be caused by external factors, e.g. changes in the law or political requirements. They can also be caused by rearrangements within an organisation itself, i.e. internal factors. We started our project creating a set of use cases that are representative for each of the change types, using cases that have their origin in real life practice. These cases are used to assess whether or not our method, which is still under development, is able to deal with the various change types as well as with their impact in a proper and flexible way. This case base will serve different purposes in different phases of development of our method and supporting tools.

One of the shortcomings of the design methods, architectures, and tools that are used in practice is that they do not help governments and organisations to keep their legal services in sync with various versions of the sources of law. Maintaining explicit links between the sources of law and the system specifications and their realisation in software components of great importance. Following those traces back is helpful when undesired or unintended results are produced, but also

when system behaviour must be justified in terms of the original sources of law in court, in parliament, or in the media.

Bench-Capon and Gordon (2009) point out the central importance of traceability between sources of law and their implementation in information systems. Attention to traceability was one of the strengths of the method developed in the E-POWER project by Van Engers et al. (2001). E-POWER implements a knowledge management solution intended to help improve the quality of legislation. We learned from E-POWER and other approaches, and recycled best practices wherever possible.

In the linked data community we also find increasing attention for provenance information about linked data. The linked data community may help help to wake up public administration, by increasing awareness of the importance of the origin of data and rules in the assessment of its applicability. For instance, Omitola et al. (2009) stress the importance of provenance information within the linked data community, but do not show recognition of the complexity of provenance in the context of complex network arrangements, typical for governments and their partners. In order to establish the meaning of data we need to be able to trace back to the legal knowledge applied to (the observation of) cases at a certain time, as the semantics of such data depends on it. Being explicit about the applied knowledge and its sources includes being explicit about versions. It requires a mature form of rule governance, which is unfortunately lacking in most large organisations today. There are however reasons to be optimistic. Currently, administrations such as the Dutch immigration service, the tax administration, and the cadastral registry are implementing rule governance based solutions. These approaches are still tentative, but they have great potential for the future.

This paper is structured as follows. First we will introduce our three layer model of reality and explain different origins of change and their effects in the three layers. Then we will explain different representation issues related to these three layers of reality. Last but not least we will explain our agent-based approach to representing the agent roles that are addressed by the legislator or come into existence when administrative processes are designed to effectuate the law. We use our agent-based approach as a knowledge acquisition instrument, rather than an operational model for building actual administrative support systems. The agent metaphor can be a powerful instrument, helping domain experts responsible for interpretation of sources of law and designing administrative processes to make better decisions about the impact of change.

1. Three Layers of Reality

One of our goals is to analyse and address the interaction between sources of law, logical representations of the meaning of these sources and business process model and legal service implementations. Distinguishing these three layers is a key factor in dealing gracefully with change.

The sources of law and the conceptualisation of their content make up two of the relevant layers. The sources of law include not only legislation and case law, but also policy statements and guidelines of lower level rule makers used in the organisation. The sources of law layer describes documents and their components, relationships between these documents, and events that happen to these documents — generally initiated by the legislator. In the next layer are the legal institutions that are represented in the sources of law, and, in a sense, created by them. Finally there is the implementation of the legal institutions in reality, in the context of Agile typically within an organisation. These three layers will be discussed in the next section.

1.1. The Three Layers

One of the goals of AGILE is to analyse and address the interaction between sources of law, logical representations of these sources of law, business process models, and public legal services, as changes will affect each of these elements individually. Conceptually, based on legal theory, we distinguish three perspectives:

1. the perspective of the sources of law and the processes that produce and change them;
2. the perspective of legal institutional reality as (re)presented by the sources of law; and
3. its implementation in brute reality.

The sources of law and the conceptualisation of their content make up two of the relevant layers. The sources of law include not only legislation and case law, but also policy statements and guidelines of lower level rule makers used in the organisation. The sources of law layer describes documents and their components, relationships between these documents, and events that happen to these documents — generally initiated by the legislator. In the next layer are the legal institutions that are represented in the sources of law, and, in a sense, created by them. Finally there is the implementation of the legal institutions in reality, in the context of AGILE, typically within a large organisation. These three layers (see Figure 1:) were already addressed by Boer and Van Engers (2010) and will be discussed in the next paragraphs.

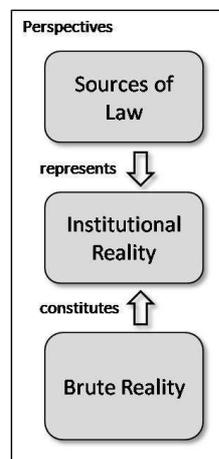


Figure 1: Layers of reality and representation.

An understanding of the notion of institution is important to apprehending the idea behind the three layers described above. Many definitions of the term institution exist, as it occurs in various areas. Generally, institutions are structures and mechanisms of social order and cooperation governing the behaviour of a set of individuals. Institutions are identified with a social purpose and permanence, transcending individual human lives and intentions, and with the creation and enforcement of rules governing cooperative human behaviour. Law is a family of institutions whose primary purpose is to create normative order by way of formalisation — i.e. a formal, institutionalised normative order (Boer, 2009) — usually wherever spontaneously arising, informal normative order fails to achieve the desired results.

The sources of law formalise the structures and mechanisms of the institution. The source of law is for our purposes writing that may be used to back an argument concerning the existence of a certain institutional entity, often, but not only, a rule, created by the legislator in a certain legal institution (Boer, 2009). The source of law is the result of a legislative act performed with the intent of creating that institutional entity, and functions as evidence of the legislative act. In this sense the sources of law are the evidence of the existence of legal institutional reality. Many other official documents have a very similar function — to function as evidence of a legal act creating some legal fact — but are written by non-legislators and do not create rules binding those other than the authors and their representatives.

The institutional reality as represented in the sources of law comes to life through the brute reality that constitutes (or counts as) it. The raising of a hand, for instance, counts as a bid in an

auction and the issuing of a document invented by some administrative agency tasked with issuing residence permits counts as an official residence permit.

It is important to understand the ramifications of the distinction when applied to rules and concepts. The institutional concepts created by the sources of law describe potential institutional entities. The rules in the sources of law do not only constrain the possible structures of institutional reality, but also constrain the mapping between institutional reality and brute reality. Any rule mapping brute reality (the raising of the hand) to institutional reality (the bid) may either be backed by the sources of law, making it a legal, institutional entity itself, or simply by knowledge of the domain, making it a non-legal implementation artefact.

The key purpose of the three layers is to distinguish problems that deal with the projection of the sources of law to institutional reality from those that deal with the projection from implementation to institutional reality. This is in itself useful, as a problem decomposition approach, and deals with the fact that solutions to these problems have different lifecycles because they are not affected by the same kinds of changes.

1.2. A classification of changes

Change may occur at all three levels of reality. Each change can be caused by an internal factor, from within the organisation, as well as by the external environment. We will describe the consequences these changes can have with respect to the other layers and use examples for illustration. Concordant to the three levels of reality we distinguish different objects of change:

1. the sources of law,
2. institutional reality,
3. brute reality,
4. the projection from sources of law to institutional reality, and
5. the projection from brute reality to institutional reality.

Probably the most recognised and addressed throughout literature is change to the contents of the sources of law. Changes in sources of law can have impact on all the things that use or refer to these sources of law. For example, brochures that explain the application of certain legislation to citizens will always have to be adjusted when these sources change in structure, which can be very costly. Law often changes due to feedback from the work floor in public administration. Sometimes the law may be changed to reflect adaptation within a single administrative organisation, and sometimes it is changed to reflect, and fix, new network arrangements between administrative organisations.

Institutional reality will generally change as a result of changes in the sources of law, but this is not at all necessarily the case, since changing legislation may simply recodify already existing institutional arrangements. In such a case, only the justifying relations between institutional entities and the sources of law change. As institutional reality changes also the representation of that institutional reality — in data structures, rules, forms, etc — will be affected. Institutional reality may also change as a result of a new interpretation of the sources of law. In this case we are dealing with change of legal conceptualisation of certain terms in existing legislation. This may happen within the boundaries of current, unchanged legislation.

Because public administration is sometimes closely involved in the legislative activity relevant to it, objectives of the legislator may also first lead to a new design for an institutional reality. In this case the sources of law are adjusted to reflect the changes proposed in the design. Last but not least, the brute reality of the organisation may change, which might require either changes in the model of institutional reality and the legislation that represents it, or changes in the mapping between institutional reality and brute reality, i.e. legal qualification. This influences the way the

rules are implemented into a specific setting. If the chosen implementation is still doing justice to the objectives, expressed by the sources of law, there is no need for changes beyond the mapping of brute reality into institutional reality. If the implementation is at odds with new rules, it has to be adapted to the new situation.

A simple example: In 2008 the Dutch immigration regulations were changed, introducing rules for knowledge migrants. The knowledge migrant has to prove his exceptional educational background. Organisations that have to work with this rule must update their systems to incorporate a check on educational background. Brute reality adapts to the new situation; how this should be accomplished is not specified. This proof may be evidence supplied by the knowledge migrant or a network arrangement with educational institutions, or both. Conversely, when the requirement for a valid passport is dropped from the law and replaced by the more general requirement that a migrant should be identified, because some refugees do not have passports, the implementing organisation may still opt to require a valid passport in a pre-existing, fast track, computer-assisted service implementation and delegate the handling of cases where no valid passport can be shown to the fully manual appeal procedure.

Another example: A law came into effect in 2002 that made it possible for a medical professional to legally assist with euthanasia. Organisations with very different perspectives, the Dutch national organisation of doctors, the public prosecutor, and the judiciary, had to search for a satisfactory solution. In this case, most doctors turned out to be unwilling to provide assistance with euthanasia. In this example institutional reality presents a service that barely works in the brute reality of many medical organisations because of lack of implementation.

A third example: In 2001 a new income tax law came into effect, which completely reconceptualised the institutional domain of income taxation. The tax administration, however, controls huge amounts of historical data about taxpayers relevant for taxation purposes. This data was collected in the context of the old law. A new meaning, and legal justification of the existence, of that data has to be found in the new law. At the same time, the new law was in many aspects merely a more transparent recodification of what were already modern views on the legal institution of income taxation.

A simple example of change caused by new technology in brute reality is the appearance of the Segway, which is now covered by traffic law. Is it a variation on an existing vehicle class, or a new kind of vehicle?

In summary, change may be initiated at all three layers, and the big maintenance question is whether changes at the other layers will follow, or whether only the projections from layer to layer are affected. An organisation may for instance first justify its existing implementation in terms of new rules, and in parallel work on an efficient implementation of the new rules.

2. Context of production and implementation

The complex problems encountered in the representation of the law can be distinguished into those of the context of production and those of the context of implementation. Context of production knowledge acquisition guidelines deal with the projection from the text of the sources of law to the rules describing the legal institutional reality presented by the sources of law. That is, between the first two layers.

Context of implementation guidelines deal with the projection from the abstract model of legal institutional reality to specific usage of rules by agents in the context of a specific knowledge environment, of specific tasks, and of specific communications with other agents. That is, between the second and third layer.

The key observation here is that a solution to a text interpretation problem that depends on the context of production of the text, does not necessarily vary with context of use of that solution. In the following paragraphs we make this distinction more concrete by connecting them to typical legal knowledge representation problems.

2.1. Context of production

To understand context of production issues, one needs to keep in mind that legislative drafters manage a very complex system. Like engineers in other domains they try to reuse components and design patterns, avoid redundant components, and coordinate their legislative actions. They also sometimes change announced plans before they have been put to effect. Context of production issues are the following:

Legislative power: It should be clear why agents accept a text as a source of law. Generally, an agent must have reason to I) accept the power of the legislator, and II) accept that the text is the outcome of the proper procedure for exercising that power. The legislative power exercised is usually the consequence of a series of delegations of legislative power for specific purposes by legislators with greater authority, originating eventually from government, parliament, and constitutional law (Boer, 2009). Internal guidelines for instance only bind employees acting on behalf of an organisation, but this often only becomes apparent by recovering the specific legislative power exercised. Legal rules may automatically become inactive if the legislative power they are based on cases to exist.

Model sentences: Legislative drafters tend to use conventionally structured sentences to achieve legal effects. Sometimes these are required by legislative drafting guidelines, and in other cases these follow from experience about the ways in which the courts tend to interpret sentences. It is realistic to expect results from automated parsing, as for instance De Maat et al. (2010) show. The functions of sentences in the sources of law are often apparent from grammatical patterns and specific keywords; These translate into design patterns for legal rules in the abstract model of institutional reality. Isomorphism: A common requirement is that there is an isomorphism between sentences in the source of law and the legal rules in the model (Bench-Capon and Gordon, 2009). This requirement may be relaxed somewhat in our view, for instance by allowing that a sentence translates into multiple legal rules because of expressive limitations of the logic used for the legal rules. Clearly undesirable, for instance, is that a sentence in the source of law translates into a condition in dozens of other rules representing other sentences, because the many-to-many reference structures that result create a greater impact on maintenance of the knowledge representation.

Internal references: References between sources of law are usually dynamic work level references, while references from knowledge representation to source of law are static expression level references (Boer, 2009, Boer et al., 2010). If a sentence refers for instance to concept C as defined in W, this concept C may change identity over the lifecycle of work W, and therefore changes the meaning of the sentence. This type of change should not require manual editing of legal rules in order to reduce the impact on maintenance of the knowledge representation. It is therefore important to distinguish static reference to expressions from dynamic references to works. Some dynamic references target future legislative discourse, for instance by referring to guidelines based on (the power delegated by) this provision.

Applicability of rules in time: Temporal applicability is a complex issue (Governatori and Rotolo, 2010, Boer et al., 2010). Distinguish when an agent applies a legal rule, when the things happened to which the rule is applied, when a legal effect follows from the application of a rule, when the (expression of the) source of law presenting the rule exists, and when that source of law is active. Complications are, for instance, retroactive and delayed applicability, ex tunc versions of sources of law, the speculative representation and application of rules that have not yet become active and are still subject to change, and retracted zombie sources of law that return from the grave at a later time. Because engineers obviously often work speculatively with sources of law that are not yet active (and because of the possibility of ex tunc versions), one should keep track of the time of actual rendering of a source of law, the *sichttag*, and the document time the rendering represents, the *stichtag*, when a sentence is translated into a rule.

Priority relationships should in principle be revealed by the sources of law, and not addressed as a logical reasoning problem on the rules (Boer, 2009). Default principles are *lex specialis*, looking

for hints in the text that one sentence is intended as an exception to the other, and *lex posterior*, giving priority to the sentence that became active (again) last. To override these principles, a legislator may explicitly declare sources of law superior over others (the *lex superior* design principle). Case law may also shed light on I) whether two rules are actually judged to be in conflict in some context, and II) which one is prior. Whether there is an actual logical conflict and defeasibility relationship between two legal rules is however not a context of production issue.

2.2. Context of implementation

When the complexities of the context of production have been untangled, and a satisfactory view of the relevant part of institutional reality in a certain time frame has been gained, another complex mapping takes place into various implementations, giving rise to context of implementation issues that must be addressed in knowledge acquisition.

Context of implementation issues all deal with situated knowledge of the law as applied in action by an agent. Context of implementation issues are the following:

Constitutiveness: What things in the world count as the things presented by the model of institutional reality (Boer, 2009). Firstly, this confronts us with the question of what things that are already there, count as something in a rule, and secondly, the law also presents the organisation with the responsibility to create things. If the organisation is for instance tasked with some formal decision, it needs to design a business process and a form, a standardised letter, for instance, in which the decision is presented.

Intentionality: The workings of legal institutional reality are not neutral. Violating the law is bad. Being eligible for a benefit is good. Legal rules are used in a goal-directed way. Goal-directed use of rules has a direct effect on the way in which they are represented in logical knowledge representation languages. The intentions of agents outside the organisation play an important role: implementation also means advertising legal procedures as services to clients, proactively sharing knowledge with clients, and taking the clients interests and expected behaviours into account in the design process.

Evidence requirements: Since formal decisions generally do not deal with things that are readily observable, the organisation needs to decide what counts as sufficient evidence for a proposition in a context (Bex et al., 2010). Does a marriage certificate for instance prove the existence of a marriage? Is there counterevidence that is even more convincing? Since evidence does not produce itself, the burden of proof (Gordon et al., 2007, Governatori and Sartor, 2010) for propositions is allocated to specific agent roles in a rule application setting. Clearly intentions play an important role: Obtaining evidence from neutral third parties is for instance preferred over obtaining proof from beneficiaries of decisions.

Defeasibility: Only in a specific setting do pairs of legal rules create a conflict for a goal-directed agent, and only on this level we can speak of defeasibility. The resolution of conflict should be consistent with the priority relationships between legal rules.

Effectiveness and efficiency: Implementation takes resources, and these must be allocated effectively and efficiently. An important consideration is whether specific types of decisions should be automated or handled manually. Hard to accept for knowledge engineers is that there are limits to the resources that can be invested into modeling. Collecting evidence for noncompliant behaviour is complex and costly, and enforcement policies are therefore usually opportunistically driven by utility and separated from primary service delivery processes. The organisation typically monitors risks, and engages in theory construction to formulate client risk profiles.

Argumentation: Intentional action should be justifiable. The organisation should be able to argue a formal decision, before and after the fact, and be able to evaluate arguments by others. Argumentation is in practice a mixture of storytelling (causal reasoning), arguing from evidence (Bex et al., 2010), and arguing about the motives of participating agents.

Given this list of issues, it is not surprising that it is hardly possible to use one meaningfully executable standard interpretation of what a legal rule means. Different implementation contexts lead to a different focus on implementation issues, and therefore to different implementations of a legal rule. In a large organisation, legal rules are usually implemented in various places in the organisation, with different purposes, and different contexts of use, to be used by different agents. The impact of many-to-many relationships from legal rules to implementation artifacts on the complexity of maintenance plays an important role. The separation of concerns between the legal institutional meaning of the legal rule and its operational meaning in implementation alleviates the maintenance problem somewhat, although legal rules will still be an input to various implementation artifacts.

3. A task-oriented perspective on rule production and implementation

The contexts issue suggests a two-step approach to rule governance. For rule governance purposes, organizations have to decide on a single, descriptive representation of the rules that does take into account the complexities of the context of production but not a context of implementation. By necessity, these rules have limited operational meaning. In implementations a derivation of those rules is used that does take into account the context of implementation.

The relevant data about the context of production comes from the sources of law itself, generally accepted constitutional and administrative legal doctrine, and, very importantly, from descriptive metadata about the sources of law. For this purpose we assume the availability of MetaLex metadata, briefly introduced in section 0. The model of institutional reality, briefly addressed in section 3.2, is based on the text and the metadata.

A central topic of this article is the context of implementation of legal rules. Knowledge of this context depends on a more general knowledge acquisition process involving domain experts. A knowledge acquisition process for public administration should effectively leverage knowledge from:

1. the legal rules as evidenced by the sources of law,
2. design experiences and design patterns used in the organisation,
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.

There are of course many solutions for dealing with context, and for the specific aspects of context distinguished in the previous section. A very flexible approach is the agent role approach, which is descriptive of the rules+actions paradigm that dominates in actual implementation of decision support solutions, and gives a natural place to argumentation, burden of proof, goal-directedness, legal positions, powers, duties, responsibilities, services, etc.

In the AGILE knowledge representation framework the notion of agent roles, introduced later in section The agent role as a knowledge acquisition device The agent role as a knowledge acquisition device, and the possibility of multi-agent system simulation, plays a central role. Agent roles are the subject of choice in simulation of complex social organisations. In the interest of leveraging existing structures and design patterns in the organisation, we merge the agent role concept with an existing generic task framework of Breuker (1994). The framework allows for robust generalised business process designs, and explains certain design patterns that we encounter in organisations. This framework will be presented in section Design patterns in implementation, before the notion of agent roles is addressed.

We extended the original generic task framework with three abstraction levels to accommodate our view of design and legislation (see Figure 2). Reference to the legal rules is in first instance a

matter of justifying action, before or after its occurrence. This fundamental principle applies regardless of whether we adopt a planning or a design perspective towards some domain regulated by law. Justification of design artifacts like business process specifications, business rules, database schemas, and law, with reference to the law, is therefore in first instance a law-based justification of design actions. In the opposite direction we can also say that design activity and legislative activity are often the result of feedback from the work floor, justifiable with reference to stories of critical incidents on the work floor.

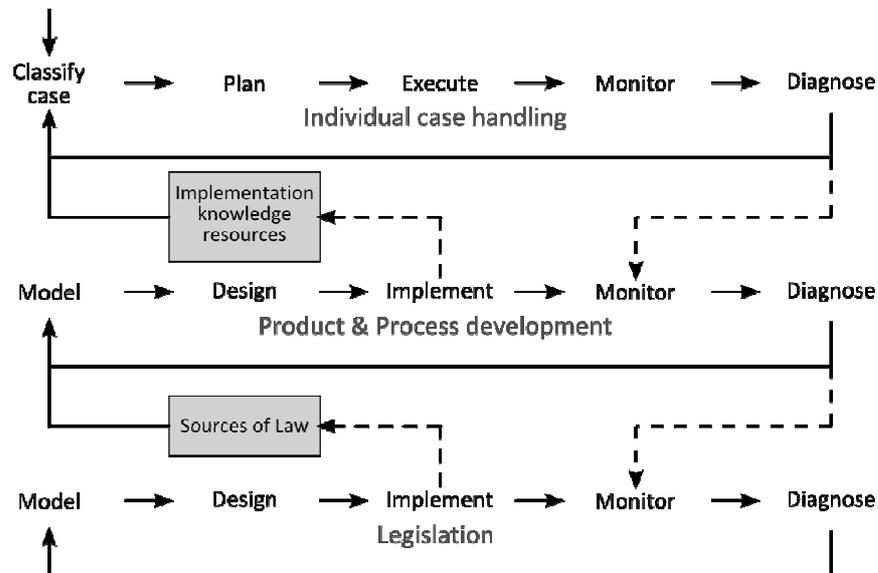


Figure 2: The case handling, development, and legislation problem solving cycles.

3.1. Design Patterns in the production of law: MetaLex

To implement traceability from knowledge representation to sources of law, the AGILE project builds on the results of our work on MetaLex XML in Boer et al. (2010).

In the MetaLex CEN/ISSS initiative, IT industry, publishers, government agencies, and academics work together to create an open XML interchange standard for sources of law and legislative resources. MetaLex serves as a common document format, processing model and metadata set for software development. MetaLex is a generic and extensible interchange framework for the XML encoding of the structure of, and metadata about, documents that function as a source of law. It aims to be jurisdiction and language-neutral, and is based on modern XML publishing concepts like a strict separation between text, markup, and metadata, building on top of structure instead of syntax, accommodation of transformation pipelines and standard APIs, and integration of Semantic Web standards. MetaLex defines mechanisms for schema extension, for adding and extracting metadata, for cross referencing, for constructing compound documents by reference, for legally sound version management, and for implementation of a naming convention. MetaLex ideally subsumes existing XML standards, requiring only minimal syntactic changes to existing XML document instances, making MetaLex compliance an unobtrusive addition to existing XML document processing and dereferencing frameworks.

For purposes of reference and self-identification, MetaLex requires adherence to an IRI-based, open, persistent, globally unique, memorable, meaningful, and – in a sense – guessable naming convention for legislative resources based on provenance information. This provenance information can be extracted in RDF form and used in Semantic Web technologies. Provenance information is described in terms of events that happened to the documents, or that the document played a role in.

In Europe, MetaLex metadata can be assumed to be freely available for Dutch, Italian, and British legislation.

The most important MetaLex design feature for our purposes, is its approach to versioning and temporal applicability. MetaLex and the MetaLex naming convention strictly distinguish the source of law as a published work from its set of expressions over time, and the expression from its various manifestations and the various locatable items that exemplify these manifestations. Another important design feature of MetaLex is the event-based view of metadata. The MetaLex standard recognises that the metadata in essence describes a context of production, which includes actions, legislative powers, and agents.

3.2. Design patterns in institutional reality

For the mapping to the model of institutional reality, we refer to Boer and Van Engers (2010) for more details. Relevant patterns in logical propositions describing the functions of legal rules revolve around the notions of *constitutiveness* and *applicability*. The context-of-implementation-neutral rules are only used for consistency checking, and only have functional significance for database integrity checking.

The legal rules represented by the source of law appeal to two separate realities, institutional reality and brute reality. They perform a mapping from brute reality, the ontological substratum, into institutional reality, the ontological superstratum. The substratum has an existence independent of the rules, while the superstratum is supervenient on the substratum and exists by virtue of social recognition of the rules of the institution. Institutional events are *constituted by* events in brute reality.

Institutional rules map out a logical space of possible models of the institution: they form the institution's ontology, and can be interpreted as terminological axioms. They are not to be considered defeasible as a matter of convention. The main function of the *constitutive rule* is to define the interface through which the state of the institution can be changed. As such, the most important question they answer is how to map brute reality *into* institutional reality. Legal rules of a constitutive nature are however not conditionals which take brute reality propositions as a premise, and institutional propositions as conclusion. Logical propositions *describing* constitutive legal rules can be categorised into *requirements*, which posit necessary conditions on the mapping between brute and institutional fact, and *indicators*, which license an inference from brute fact to institutional fact, if and as long as it is consistent to do so.

Applicability plays a central role as soon as the logical proposition and the legal rule are separated. If a logical proposition states that a certain proposition indicates or entails another proposition, we must add to this conclusion that a legal rule has been applied. Any logical proposition about the application of a legal rule must conclude that the legal rule has been applied. A special class of legal rules (generally described as requirements) constrains the applicability of other rules. *Applicability rules* are used to avoid repetition of the same requirement, to demarcate the extent of jurisdiction over people, territory, and substance, and applicability in time, and to restrict the applicability of legal rules made by a lower legislator. The *choice rule* makes the application of one legal rule conditional on the application of another legal rule, if the combined application of two legal rules is judged to be problematic. Applicability rules can be used as a guide for clustering sources of law into coherent domains of application, and are generally used for that purpose by the law.

The context-neutral representation is not based on an analysis of legal positions. This requires a context of implementation, and is addressed in section The agent role as a knowledge acquisition device.

3.3. Design patterns in implementation

Organisations, like people, and perhaps even more than people, are routine decision makers. It therefore makes sense to take a generic framework for problem solving as our starting point. Breuker (1994), some two decades ago, presented a typology of problems and views on problem-solving. This typology, the suite of problem types, was based on an analysis of the problem and task decompositions found in knowledge-based system literature. Important theses of Breuker (1994) were 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 typical 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, as we will show in the next section.

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 an intelligent agent feels able to control a domain by encapsulating processes into fixed structures, it decides on a design for dealing with a type of problem and implement it. When it feels this is not feasible, given the characteristics of the domain or problem, it makes 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. 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.

Boer and Van Engers (2010a) proposed a classification of types of monitoring and diagnosis processes which will not be repeated here, starting with 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 conceptualisations: the organisation and its environment as a normative system and ontologically coherent institution, and the organisation as a goal directed agent in an environment. In the context of monitoring experience-based agent profiles are used.

Figure 2 also expresses an additional observation we made, beyond the suite of problem types presented by Breuker (1994). 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 Breuker (1994). 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 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.

Figure 2 reflects our interpretation of design and policy making as core activities of the organisation. Public administration and policy makers are mutually dependent; Public administration is in charge of efficient operationalisation, 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 reorganisation, 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 are clear possibilities for monitor-based automated resource scheduling and business process realignments.

4. A business process example

In this section we will explain the application of generic problem types in the context of business processes. Design activities in public administration are obviously not based on interpretation of the law alone. The organisation 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 generic tasks or service design patterns in organisations, that are replicated in different places with different legal content.

Figure 3 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 generalisation, workflow coordination is of course not entirely arbitrary from a legal point of view. In some cases the organisation is for instance obliged to collect evidence or obliged to enforce, and this limits the possible workflow coordination options. This is the design compliance question. We believe design compliance is a lesser and better understood knowledge management problem in public administration: design compliance does not address accountability, robustness, and agility explicitly.

In Figure 3 we have, rather arbitrarily, chosen for a basic distribution of tasks over agent roles D for desk, H for request handler, R for risk assessment desk, 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 Figure 3. It represents a generalised problem solving paradigm.

The second common invariant structure is the background diagnostic process behind any service delivery process that fits available evidence to noncompliance storylines, 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 organisational resources in comparison with primary service delivery processes, but they are often less well described from a knowledge management point of view.

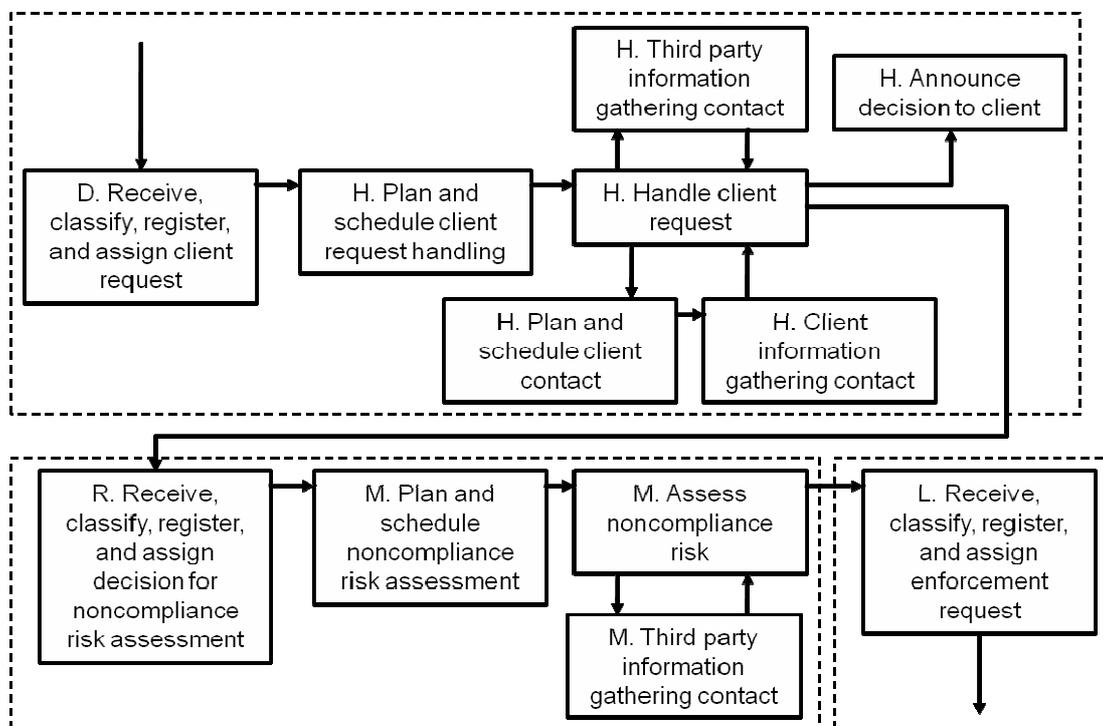


Figure 3: Repetition of the same generic task design patterns in business process specifications.

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 organisation may however also decide to schedule a monitoring process to verify that it does not fit the storylines for a sham marriage, in which case it may enforce. This time frame for this monitoring process completely dissociates it from the service delivery process.

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 internalisations of a legal rule, for instance an obligation, in different tasks and different agent roles, and clearly delineates the different knowledge contexts in which the legal rule is applied.

The opportunistic aspect of many real world compliance monitoring problems for instance focuses attention to noncompliance storylines 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 missing information. From this point of view, developments in administrative case law can also be understood well. An externalised norms approach to compliance trivialises the detection of violations and does not address these issues at all.

The business processes presented reflect the problem solving framework of figure 2 quite well. The processes reflect actual reusable service components of one of the organisations with us in the AGILE project. We do not believe, however, that the business process is the best way to create reusable components representing law-based services.

5. The agent role as a knowledge acquisition device

To deal with the complexities of context of implementation, the concept of the agent role as a knowledge acquisition device plays a central role in the Agile project. Other objects of interest, for instance the action, business process, situation, etc, are also candidates for that role, but the agent role is eminently suitable for this purpose, as we will explain.

Agent roles, for instance buyer, seller, real estate agent, and notary lawyer, are associated with a set of abilities, and a set of susceptibilities to actions of others, and with goals, plans, and beliefs typical of that role (Masolo et al., 2004, Hoekstra et al., 2008, Boer, 2009). One can interpret it as an internal description of what moves that agent, if we attribute the agent role to it, but also as a description of the affordances of an environment from the perspective of that agent. Abstractly, the environment of a seller for instance consists of buyers, who can offer, accept, and pay, and the seller is able to offer, accept, and deliver, and normally has the goal of selling something, delivering it, and receiving payment.

This perspective is influential in legal theory. The principal aim of Hohfeld's work (Sartor, 2006), for instance, which is more or less a standard view on automated normative reasoning, was to clarify jural relationships between parties. Hohfeld asserts that there are eight such entities: right, privilege, power, and immunity along with their respective correlates of duty, no-right, liability, and disability. A right of an agent towards another agent is for instance equivalent to a duty of that other agent towards the first, etc. Hohfeld's approach makes hidden assumptions about agent roles explicit. Hohfeld's analysis distinguishes the ability to act, or power, and thereby to cause a change of position, and the goal of causing a certain change of position, or duty, and most importantly, the one who acts and the one who expects actions of the other.

Implementation of agent roles as multi-agent system agents (instead of as constituents of the agents) focuses attention on important aspects of the implementation of the law, while de-emphasising less important aspects:

1. Organisational knowledge about agent behaviour is associated to the stereotypical roles agents play, rather than the individual agents themselves.
2. Representations of legal rules often naturally address an agent role, in the kind of settings with which we are dealing .
3. Representation of agent roles de-emphasises the distinction between strategic management of agent roles inside an agent and coordination between a number of agents with distinctive roles in a group. Both can be instances of the same behaviour pattern on different scales.
4. Representation of agent roles de-emphasises classical multi-agent system agent requirements like autonomy, proactiveness, and intelligence, which tend to lead to complex implementations, and instead emphasises straightforward, unambiguous, and transparent agent belief sets, plans, and goals.

Each agent role represents a subjective perspective on a domain, associated with a set of abilities and plans for that domain. A notable weakness in public administration, at least in rule governance terms, is leveraging experience about the behaviour of agents in the domain. This weakness is most evident in the absence of explicit record keeping of theory construction about noncompliance in the design phase, even though experience about critical incidents in the past and the ability to predict future incidents is usually available in the organisation and used in design. This weakness is to a large extent simply a consequence of not taking perspectives during modeling.

5.1. An example: the seller

In this example we bring together the agent role as the addressee of legislation, the organisation, and its parts, as role playing agents engaged in generalised problem solving behaviour, and the client noncompliance storyline as an agent role. The noncompliance scenario underlines the importance of modeling from an agent role perspective instead of simply modeling an environment in which agent roles are participants.

We assume that the reader of this paper understands the process of buying, and knows of different procedural variants of this process. The law says that 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 promptly pay the price to the seller, and a duty of the seller to promptly 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 cannot be complied with.

Let us assume we are designing the business process of the seller. 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. The presentation of generic tasks suggests we have to distinguish service delivery and monitoring functions. The organisation has to attend to the following goal/event-condition-action statements to describe the seller agent role as a legal abstraction:

1. If the seller has an interest in a sale, then the seller has the intention to perform an offer, and then to monitor acceptance, and to monitor the sale;
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 perform an acceptance, and to monitor the sale;
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 perform delivery.

That service delivery and monitoring are best conceived of as different, concurrent processes will be apparent, because many reasonable serialisations of workflow will otherwise lead to a deadlock. If the buyer for instance requires delivery before paying, and the seller requires payment before delivery, no sale can come about. When one of the parties is a natural person, organisations expect that they will show the flexibility to conform to the procedure the organisation decides on,

but it is a fact that some organisations cannot buy from some sellers because bureaucratic procedures make the transaction impossible.

The sales example provides us with examples of regulative and constitutive function monitoring, as distinguished by Boer and Van Engers (2010a):

- 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 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 we will see later.

The generalised business process design of section A business process example work well for the sales example. 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 organisation 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.

In keeping with Hohfeld's analysis, we have thus far tacitly assumed that only buyers are interested in monitoring sellers, and vice versa. The use of agent role-based representation for noncompliance storylines allows for more interesting variants.

Common sense dictates that normally seller and buyer agents are played by different players with complementary interests, but there are in fact a number of tax evasion schemes that assume that an agent knows that he is selling to himself, but expects 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 important client. Many variations of this scheme exist, and the abstract scheme has survived many changes of the law. The issue with sellers who sell to themselves is not even the identity issue as such, but the uneconomic prices set for the property exchanged. The buyer-seller has the wrong goal, tax evasion, in mind when he determines the price of the property. This places the buyer-seller in a broader category that includes kickback and extortion schemes, all characterised by prices set at uneconomic levels. The nature of the relationship between agents, and the direction of the deviation from the expected price level helps us determine which scenario we could be dealing with.

The tax evasion scheme gives a completely different background to the meaning of legal rules about buying and selling. It is also an example of noncompliance knowledge in large organisations, and it shows how the same scheme can be used by one person (who is both buyer and seller) or a groups of persons with aligned interests (in tax evasion). Finally, note that tax evasion cannot be usefully connected to a business process in the organisation, since it is all about the perspective of the other agent, external to the organisation.

6. Conclusion

Coping with change can be a serious challenge to governments and citizens. In this paper we tried to unravel various aspects of change using a layered model of reality as a basis. We perspectives on change based on the three layers and the interactions between those layers, resulting in five types of change. We then described some typical knowledge representation aspects in the context of production and the context of implementation which make it hard to straightforwardly translate a text into a formal model. We then introduced a generic problem solving

task model which originates from Breuker (1994) and showed how this generic task model fits onto the context of production and the context of implementation.

This generic task model also covers the production process of sources of law. The metadata part of the current de facto standard for sources of law, MetaLex, contains traces of these tasks, taking an event-based approach. We also showed how the generic task model fits onto a generic business service architecture that can be used to describe or design many public administrative processes. In both cases however, the assignment of agents responsible for task execution is left underexposed. We showed that making the agents and their agent roles explicit can help us to better understand the institutional situation, degrees of freedom in design, and impact of change. By taking into account the perspectives of the agents involved, and by making explicit their intentions, beliefs, desires, one gains an understanding of how regulations and their implementation affects these agents, also in case of noncompliant behaviour.

With the design methodology that we are developing in AGILE we aim to support effectuation of regulations, taking into account issues such as reference, and representation (from MetaLex), applicability, and constitutiveness. In our approach we make a clear distinction between abstract institutional entities and their implementation as entities in organisational reality.

An additional challenge is to account for the relation between jural relationships on the one hand, and agents, goals, and services on the other hand. This relation is to be found in the representation of abilities and plans typical of agent roles.

The clear distinction we make between layers helps us to more precisely address the impact of changes of law in the representation system, and indirectly in the specification of legal support systems. These distinctions are even more useful if we aim at supporting legislative drafters and policy makers, who need to predict the outcomes of their intended changes not only with respect to the legal outcomes of the categories of cases determined by the sources of law, but also with respect to (human) resource consumption in public administration, incentives for noncompliance, etc.

Although the AGILE project is addressing an important problem that has increased with the growing dependency on supporting legal information systems and complex systems of rules, our scope is not limited to that. We hope to develop a transparent metamodel of the legal system. The combination of this legal theory with economic models may help to improve reasoning about the effectiveness of law, which is relevant both from a scientific as well as a pragmatic perspective. This theory will not only be beneficial to governmental organisations that have to cope with change. It will also help citizens and their representatives to better analyze and address the problems they have with the law and help them to come up with better solutions to those problems, contributing directly to more effective governance of our society.

References

- Bench-Capon, T. and Gordon, T. F. (2009). Isomorphism and argumentation. In Proceedings of the 12th International Conference on Artificial Intelligence and Law, pages 11-20, New York, NY, USA. ACM.
- Bex, F., van Koppen, P., Prakken, H., and Verheij, B. (2010). A hybrid formal theory of arguments, stories and criminal evidence. *Artificial Intelligence and Law*, 18:123-152. 10.1007/s10506-010-9092-x.
- Boer, A. (2009). *Legal Theory, Sources of Law, & the Semantic Web*. *Frontiers in Artificial Intelligence and Applications* 195. IOS Press, Amsterdam, the Netherlands.
- Boer, A., and Van Engers, T. (2010). Knowledge Acquisition in Public Administration. In Proc. Of the European Knowledge Acquisition Workshop 2010. *Lecture Notes in Computer Science*, Springer Verlag .
- Boer, A. and van Engers, T. (2010a). Generic problem solving tasks and agent roles in public administration. In Winkels, R., editor, *Legal Knowledge and Information Systems. Jurix 2010: The Twenty-third Annual Conference*, *Frontiers of Artificial Intelligence*. IOS Press.
- Boer, A., Hoekstra, R., de Maat, E., Vitali, F., Palmirani, M., and Ratai, B. (2010). *Metalex (open xml interchange format for legal and legislative resources)*. Technical Report CWA 15710:2010 E, European committee for standardization (CEN), Brussels.

- Breuker, J. (1994). Components of problem solving and types of problems. In Steels, L., Schreiber, G., and Van de Velde, W., editors, *A Future for Knowledge Acquisition*, volume 867 of *Lecture Notes in Computer Science*, pages 118-136. Springer Berlin / Heidelberg.
- De Maat, E., Krabben, K., and Winkels, R. (2010). Machine Learning versus Knowledge Based Classification of Legal Texts. In Winkels, R., editor, *Legal Knowledge and Information Systems. Jurix 2010: The Twenty-Third Annual Conference, Frontiers of Artificial Intelligence*. IOS Press.
- Gordon, T. F., Prakken, H., and Walton, D. (2007). The Carneades model of argument and burden of proof. *Artificial Intelligence*, 171(10-15):875--896. *Argumentation in Artificial Intelligence*.
- Governatori, G. and Rotolo, A. (2010). Changing legal systems: legal abrogations and annulments in defeasible logic. *Logic Journal of IGPL*, 18(1):157-194.
- Governatori, G. and Sartor, G. (2010). Burdens of proof in monological argumentation. In Winkels, R., editor, *Legal Knowledge and Information Systems. Jurix 2010: The Twenty-Third Annual Conference, Frontiers of Artificial Intelligence*. IOS Press.
- Hoekstra, R., Breuker, J., Di Bello, M., and Boer, A. (2008). LKIF Core: Principled ontology development for the legal domain. In Breuker, J., Casanovas, P., Klein, M., and Francesconi, E., editors, *Law, Ontologies, and the Semantic Web*. IOS Press.
- Masolo, C., Vieu, L., Bottazzi, E., Catenacci, C., Ferrario, R., Gangemi, A., and Guarino, N. (2004). Social roles and their descriptions. In *Proceedings of Knowledge Representation Workshop*.
- Omitola, T., Gibbins, N., Shadbolt, N. (2009). Provenance in Linked Data Integration, in *Proceedings of the Linked Data On the Web 2009*. C. Bizer, T. Heath, T. Berners-Lee, K. Idehen (eds), CEUR workshop proceedings, ISSN 1613-0073.
- Sartor, G. (2006). Fundamental legal concepts: A formal and teleological characterisation. *Artificial Intelligence and Law*, 14(1-2):101-142.
- Van Engers, T., Gerrits, R., Boekenoogen, M., Glassée, E., and Kordelaar, P. (2001). POWER: Using UML/OCL for modeling legislation - an application report. In *Proceedings of the International Conference on Artificial Intelligence and Law*. ACM Press, 2001. 1-58113-368-5/01/0005.

About the Authors

Tom van Engers

Van Engers (1961) studied Cognitive Artificial Intelligence at the Utrecht University and received his PhD at the faculty Mathematics and Information Sciences at the Free University in Amsterdam. He has worked at the Ministry of Finance in different functions amongst them as research manager. Van Engers is professor in Legal Knowledge Management at the University of Amsterdam and managing director of the Leibniz Center for Law. Van Engers coordinated several international research projects such as E-Power, Trias Telematica, Estrella and SEAL (see for more information the Leibniz Center for Law website: www.LeibnizCenter.org). His research is particularly focused on Rule Governance and Legal Engineering in which artificial intelligence and design methods from computer sciences are combined with legal reasoning.

Alexander Boer

Alexander is a senior researcher of the Leibniz Center for Law at the University of Amsterdam. His research interests include computational legal theory, legal ontology, e-Government, and generally the development of innovative concepts for the application of IT in the field of law. Dr. Boer worked on MetaLex XML, and chaired the technical committee of the MetaLex CEN/ISSS workshop for several years. He received his MA degree in social science informatics, and his PhD degree in Computational Legal Theory (Law) from the University of Amsterdam.